



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

“Caneta Ótica para registro e contabilização automática de produtos”

Autor:

LUCIANO CORTEZ TOLEDO
2003724/0

Prof. M.c. Maria Marony Sousa Farias Nascimento
Orientadora

Brasília-DF, novembro de 2008.



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

“Caneta Ótica para registro e contabilização automática de produtos”

Brasília-DF, novembro de 2008.

Agradecimentos

Agradeço a Deus pelas graças concedidas e possibilidades de aprendizado, consubstanciados pela saúde e especialmente pela minha maravilhosa família.

Agradeço em especial à minha mãe, que proporcionou esta realização, por todo o amor incondicional, suporte, dedicação, incentivo e por acreditar em minha capacidade. Sem ela, este projeto de vida seria praticamente impossível.

Ao meu pai, por todo o amor e incentivo.

À minha esposa Ana Cristina e meu filho Igor, por todo carinho, amor, incentivo e dedicação, e por sempre me apoiarem em todas as horas e decisões.

Ao meu irmão Ricardo, pelo incentivo, apoio e palavras de sabedoria, assim como as horas de estudo que passamos juntos.

À minha tia Zara e minha avó leda, por acreditar e me apoiar mesmo nos momentos incrédulos.

À Mestre Prof^a Maria Marony, pela dedicação, atenção e esforço em me orientar, e ao Mestre Prof^o Francisco Javier, por todo apoio e incentivo.

Resumo

O objetivo deste trabalho é confeccionar um instrumento, uma caneta ótica, capaz de registrar e contabilizar automaticamente o nome e valor de produtos, via código de barras, enquanto simultaneamente exibe-os na tela da caneta, bem como o somatório de todos os valores lidos. Ao término das compras, o instrumento é capaz de transferir os dados registrados para o computador central e ato contínuo o cliente recebe a fatura.

O programa desenvolvido decodifica a leitura do código de barras e extrai o valor da compra para a porta USB que é acoplada à caneta ótica. O arquivo é extraído e criptografado para evitar fraudes. O projeto foi planejado e desenvolvido com essas ferramentas, em conformidade com o padrão do processo operacional. O protótipo foi construído e testado de maneira sistemática e racional, o que permite alcançar os objetivos propostos.

Palavras-chave: informática, caneta ótica, automatização de compras, políticas de atendimento ao cliente.

Abstract

The objective of this study is to bring out the possibilities of a device, optical scanners, which is able of reading a barcode and automatically register the name of the product and its price. Simultaneously it computes the total amount of these registered values, whilst it displays in a built-in screen the last recorded variables. At the cashier the device is able to transfer all recorded data to the main frame, while the customer receives the bill without delay.

The developed program decodes and transfers the total sum of values to a USB port, which is assembled to the optical scanners. The data are recovered and a cryptogram is made to avoid fraud. The project is planned and developed using these tools according to standard procedures. A prototype is developed and tested in a systematic and rational way, in order to accomplish the objectives.

Key words: information systems, optical scanners, purchase automatization, customer policy.

SUMÁRIO

Lista de Figuras	VII
Lista de Tabelas	IX
Lista de Abreviaturas	X
Capítulo 1. INTRODUÇÃO	1
1.1 Motivação	2
1.2 Hipótese	2
1.3 Objetivo Geral	3
1.4 Objetivos Específicos	3
1.4.1 Teórica - Objetivos específicos	3
1.4.2 Prática – Objetivos específicos	3
1.5 Metas	4
1.6 Organização	5
1.7 Etapas da elaboração e desenvolvimento do projeto	6
Capítulo 2. APRESENTAÇÃO DO PROBLEMA	7
2.1 Encaminhamento de Solução	8
2.2 Conseqüências do Problema	10
2.3 Fatores Identificados	10
Capítulo 3. REFERENCIAL TEÓRICO E BASES METODOLÓGICAS	12
3.1 Dispositivos	13
3.1.1 Microcontroladores	14
3.1.2 Display LCD	17
3.1.3 A Tecnologia Microcontrolada	19
3.1.3.1 O Microcontrolador PIC 16F648A	21
3.2 Leitor de Código de Barras (Barcode Scanner)	23
3.3 Código de Barras – Conceitos Básicos	24
Capítulo 4. A PROPOSTA DO MODELO (PROTÓTIPO)	36
4.1 Explicação Detalhada do Código Fonte	47
4.1.1 Diagramação Gráfica	47
4.1.2 Principais Funções	48
4.2 Testes Utilizando a Placa Definitiva da Caneta Ótica	64
Capítulo 5. APLICAÇÃO DA SOLUÇÃO	69
5.1 Laboratório de Experimentação	69
5.2 Resultados Obtidos	70
5.3 Variáveis de Impacto Indireto no Projeto	71
5.4 Resultados Obtidos das Comparações Efetuadas	71
Capítulo 6. CONCLUSÃO	73
6.1 Sugestões para Trabalhos Futuros	74
REFERÊNCIAS	76
Apêndice I	78
Apêndice II	81
Apêndice III	84

Lista de Figuras

Figura 1.1 – Ilustração de display em LCD acoplado numa Caneta Ótica.....	4
Figura 1.2 – Etapas da Elaboração do projeto	6
Figura 3.1 – Imagem de um Microcontrolador de aparelhos domésticos	15
Figura 3.2 – PIC 16F84	15
Figura 3.3 – PIC BASIC Step 1	16
Figura 3.4 – PenBS – Driver para Pen Drive	17
Figura 3.5 – Display LCD	19
Figura 3.6 – Microcontrolador 16F648A	21
Figura 3.7 – Pinagem do PIC 16F628A	22
Figura 3.8 – Modo de operação do oscilador do PIC 16F648A	23
Figura 3.9 – Leitor de código de barras portátil	24
Figura 3.10 – Codificação do dígito 0	26
Figura 3.11 – Codificação dos dígito 0 a 4.....	26
Figura 3.12 – Codificação dos dígito 5 a 9	27
Figura 3.13 – Codificação do numero 1998	27
Figura 3.14 – Exemplo de um GS1-128	29
Figura 4.1 – Kit de testes de funções de dispositivos	36
Figura 4.2 – Código de programa de teste de leds da placa de testes no CompiladorMikroc	37
Figura 4.3 – Aplicativo IC-Prog v1.05D	38
Figura 4.4 – Caneta Ótica conectada na porta serial da placa de testes Acepic.....	40
Figura 4.5 – Programa versão 2.8 compilado no MikroC	42
Figura 4.6 – Programa aberto na versão 2.8 no aplicativo IC-Prog a ser gravado no PIC 16F648A	43
Figura 4.7 – Placa de desenvolvimento com o programa na versão 2.8 gravado e executado	44
Figura 4.8 – Representação do código de barras dos caracteres 3D	45
Figura 4.9 – Leitura do código de barras do produto “feijão”	46
Figura 4.10 – Leitura do código de barras do produto “amaciante”.....	47
Figura 4.11 – Máquina de estados do programa da Caneta Ótica.....	48
Figura 4.12 – Tela de inicialização da Caneta ótica	49

Figura 4.13 – Tela de extração dos dados para o pen drive	52
Figura 4.14 – Comparação entre execução do software no hardware	53
Figura 4.15 – Função “Apagar da lista compras” acionada	55
Figura 4.16 – Código agindo no hardware (efetuando leituras dos códigos de barras dos produtos)	56
Figura 4.17 – Código do programa na versão 2.8 em transformado em hexadecimal pelo aplicativo IC-Prog	58
Figura 4.18 – Extração de dados da caneta ótica	59
Figura 4.19 – Lay-Out da placa definitiva do projeto Caneta Ótica	60
Figura 4.20 – Imagem da placa definitiva da solução Caneta Ótica	62
Figura 4.21 – Placa definitiva com placa PenBS e pen drive	63
Figura 4.22 – Teste de leitura de código de barras do produto “açúcar”	64
Figura 4.23 – Teste de leitura de código de barras do produto “macarrão	65
Figura 4.24 – Extração de dados do sistema para o pen drive	66
Figura 4.25 – Extração de dados do sistema para o pen drive.....	67
Figura 4.26 – Lista dos produtos lidos pela caneta ótica	68
Figura 5.1 – Leitura do código de barras do produto “feijão”.....	69
Figura 5.2 – Leitura do código de barras do produto “sal”	70
Figura Apêndice 1.1 – Cronograma do projeto. Modo planilha	79
Figura Apêndice 1.2 – Cronograma do projeto. Modo Gráfico de Gannt.....	80
Figura Apêndice 2.1 – Esquema elétrico placa de desenvolvimento.....	82
Figura Apêndice 2.2 – Esquema elétrico placa de desenvolvimento.....	83

Lista de Tabelas

Tabela 1 – Intervalos médios entre chegadas e tempos médios de serviço dos consumidores nos caixas	11
Tabela 2 – Módulos LCD disponíveis	18
Tabela 3 – Construção de uma seta para cima	20
Tabela 4 –Identificadores de Aplicação – cód. de barras tipo GS1-128	30

Lista de Abreviaturas

AAMMDD	Sigla para convenção de data (Ano; Mês; Dia)
ABAC	Associação Brasileira de Automação Comercial
ABNT	Associação Brasileira de Normas Técnicas
AC	Alternate Current
Bps	Bits Por Second – bits por Segundo
CGRAM	Character Generator RAM - RAM do Gerador de Caracteres
CGROM	Character Generator ROM – ROM do Gerador de Caracteres
DDRAM	Data Display RAM de dados do display
EAN	European Article Number. (Internacional Article Numbering Association).
EEPROM	Electrically Erasable Programmable Read Memory
E/S	Entrada/Saída
Hz	Hertz
I/O	Input/Output
LCD	Liquid Crystal Display – Display de Cristal Líquido
LED	Light Emitter Diode – Diodo emissor de luz
kHz	Kilohertz
mA	miliampér
Mbps	megabits por second – megabites por segundo
MCU`s	Microcontroladores
MHz	Megahertz
PC	Personal Computer – Computador pessoal
PIC	Peripheral Interface Controller
RAM	Read Access Memory – Memória de Acesso Aleatório
ROM	Read Only Memory – Memória de apenas leitura
USB	Universal Serial Bus – Porta de serial universal
V	Volts

Capítulo 1. INTRODUÇÃO

O crescimento populacional das cidades tem aumentado consideravelmente, causando transtornos em muitas atividades do dia-a-dia. Um exemplo disso é quando se vai ao supermercado e encontram-se extensas filas de clientes aguardando sua vez para passar suas compras no caixa. O excessivo número de pessoas aguardando a ação de desmagnetização de seus produtos escolhidos, assim como contabilização do valor total e pagamento do mesmo, faz deste processo uma espera estressante, além da demora no ato de computar cada item da compra no sistema do caixa. Em resumo, essa seqüência tem efeito negativo que se reflete no cliente, nos colaboradores e no estabelecimento comercial.

O ato de ir a supermercados e similares é uma das necessidades da vida moderna, e esse ato deve ser aprazível ao invés de estressante. Longas filas nos caixas fazem com que as pessoas tentem alternativas, como efetuar as compras pela internet, com os produtos sendo entregues em domicílio. Mas, esta é apenas uma alternativa para poucos, pois ainda não satisfaz por completo muitos clientes. Para aqueles que podem pagar um valor mais alto em suas compras, no custo total deste tipo de benefício está embutida a cobrança da entrega em domicílio. Ou seja, não é uma solução para a maioria dos consumidores.

Algumas empresas tentaram solucionar este tipo de problema com o uso de equipamentos modernos, mas muito caros para a economia atual. Essas soluções são de difícil implantação, por conta da inadequação dos estabelecimentos comerciais e também pela cultura dos próprios clientes. Considerando as leis do mercado, aqueles estabelecimentos se organizam de acordo com seus custos e benefícios. Assim, não disponibilizam altas quantias para a solução de tais problemas. Para esta situação é importante encontrar uma solução compatível com o custo e benefício da atual economia, e que seja acessível ao orçamento das empresas.

Há necessidade premente de aumentar a satisfação de todos aqueles que convivem em ambiente de vendas a varejo, e ampliar o fluxo de consumidores na área física do estabelecimento. É sobre isto que este projeto trata: trazer uma solução para minimizar o problema do excesso de filas em supermercados, com a

utilização de um novo equipamento que faz a leitura ótica do produto no momento da compra, e a contabilização imediata do total gasto.

1.1 Motivação

A motivação para o desenvolvimento deste trabalho foi à idéia de gerar uma solução que minimizasse o problema da esperas em filas dos supermercados e em comércios varejistas em geral, agilizando não só o atendimento ao cliente, como também facilitando o trabalho do fornecedor.

Ainda não houve uma solução compatível com o mercado, que gerasse baixo custo e alto benefício para esses grupos de estabelecimentos comerciais. Todas as soluções até hoje desenvolvidas geraram despesa superior ao orçamento disponível, e por isso não entraram em produção. [IBM 1]

Solucionar um processo estafante sempre é visto como benvinda pela população, pois é interessante notar que os consumidores correlacionam o tempo total gasto no supermercado com a falta de organização geral do estabelecimento. Conforme MORABITO e LIMA (2000) surgem então um *tradeoff*¹ a ser analisado entre o nível de serviço a ser oferecido ao cliente e a escolha de capacidade do sistema de atendimento. [ABE 2]

1.2 Hipótese

Se existem dificuldades operacionais para efetivação final da compra em ambiente de varejo com grande demanda, então é possível utilizar um equipamento que incorpore um software para a solução do problema. Este equipamento deve ser capaz de registrar, qualificar e quantificar o produto no momento da compra e agilizar o processo de pagamento.

¹ Expressão que define uma situação em que há conflito de escolha.

1.3 Objetivo Geral

O objetivo principal do projeto é construir uma caneta ótica que anote automaticamente os preços dos produtos, via código de barras, e totalize o valor da compra na própria caneta, permitindo ao cliente ter a exata noção de seu gasto durante o processo de compras, além de permitir ao caixa do estabelecimento maior agilidade para repassar o valor das compras para o computador central, e registrar seu pagamento.

1.4 Objetivos Específicos

Este trabalho está dividido em duas abordagens: teórica e prática.

1.4.1 Teórica - Objetivos específicos

- Apresentar o problema-base e a sua respectiva solução;
- Apresentar as metodologias e técnicas de desenvolvimento a serem utilizadas para o desenvolvimento do produto-solução;

1.4.2 Prática – Objetivos específicos

- Adaptar uma caneta ótica padrão de mercado (com leitura laser SERIAL) com Microcontrolador PIC16F468A, visor de celular para saída gráfica e porta USB para leitura e comunicação com pen-drives;
- Criar um software que seja capaz de se comunicar com a caneta ótica, realizar uma leitura de código de barras, armazenar os dados de entrada no chip, apresentar o produto lido e seu respectivo preço no visor e exportar os dados armazenados para uma pen-drive;

1.6 Organização

Esta monografia está organizada em seis capítulos:

1. O primeiro capítulo traz a colocação do problema e enuncia a hipótese, além de definir os objetivos e metas.
2. No capítulo dois é apresentado o problema a ser solucionado, dando ênfase aos benefícios da utilização do modelo proposto no projeto, em prol da resolução dos problemas citados. Enfatiza também como o problema está sendo tratado, quais as soluções existentes, as conseqüências e efeitos negativos pela persistência do problema, além de destacar os benefícios da solução proposta.
3. No capítulo três são analisadas as abordagens e técnicas discutidas no capítulo anterior, em especial, sobre os componentes eletrônicos a serem pesquisados e projetados, as devidas explanações sobre o referencial teórico e bases metodológicas.
4. No capítulo quatro é mostrado o desenvolvimento da solução proposta (o objetivo principal deste trabalho), técnicas de confecção/produção da caneta ótica, componentes eletrônicos utilizados, com as devidas explicações correlatas.
5. No capítulo cinco é demonstrado os testes efetuados com a solução proposta, testes de leituras de código de barras e visualização de valores no display LCD. Compara também as médias de tempo gasto com o sistema atual para com o sistema proposto (solução caneta ótica contabilizadora automática de produtos).
6. O último capítulo traz as conclusões decorrentes de uma análise do trabalho realizado, e de como a metodologia utilizada pode ser útil em outros projetos correlatos. E também cita propostas para projetos futuros em continuidade deste.

1.7 Etapas da elaboração e desenvolvimento do projeto

A execução do plano acompanha uma sequência hierárquica de atividades, até a confecção da caneta ótica e testes que comprovam sua eficácia.

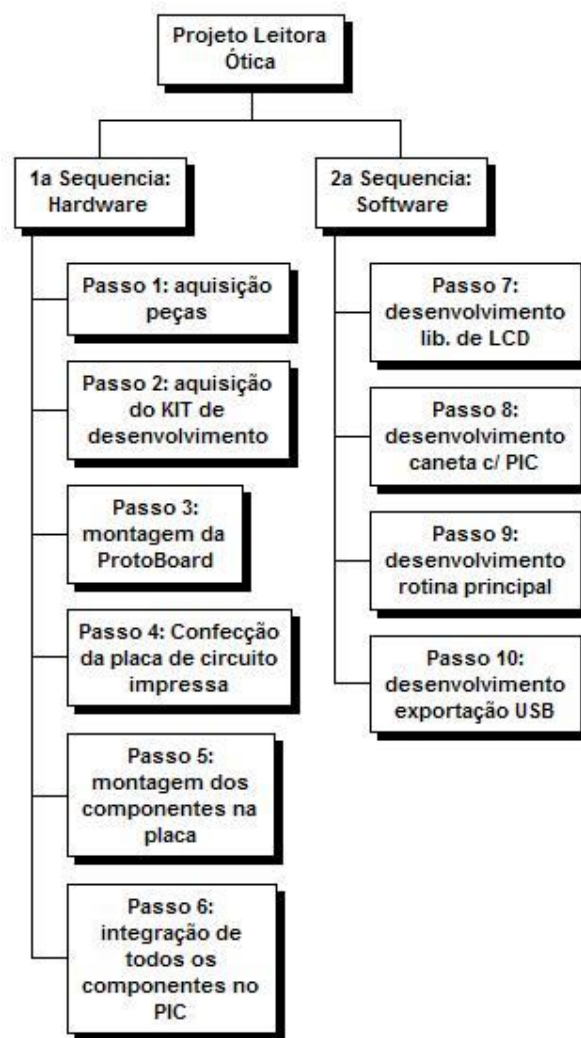


Figura: 1.2 – Etapas da elaboração do projeto. Fonte: Do Autor

Capítulo 2. APRESENTAÇÃO DO PROBLEMA

Enumerando-se os problemas encontrados quando se vai às compras em supermercados e outros estabelecimentos, têm-se: 1- formar fila nos caixas; 2- remover todos os produtos do carrinho e colocá-los na esteira do caixa; 3- acomodar todos os produtos em cima da esteira; 4- demora do caixa em passar os produtos no leitor de código de barras; 5- ensacar todos os produtos retirados do carrinho de compras para contabilização; 6- colocar de volta no carrinho todos os produtos 7- perda de tempo gasto com todas estas operações.

Os supermercados e outros estabelecimentos comerciais similares apresentam-se praticamente com a mesma estrutura. Os leitores de código de barras são acoplados ao equipamento do caixa, ligados por comunicação de rede ou serial às registradoras computadorizadas. Ainda é utilizado o sistema de esteira no caixa, onde se inicia um longo processo de passos até o pagamento das compras pelo cliente.

Para aumentar o volume de vendas, gerentes de supermercados têm se preocupado em melhorar o nível de serviço ao consumidor. De acordo com uma pesquisa realizada em 1996, na grande São Paulo (OMC/EAN, 1996), o consumidor em geral é fiel ao supermercado que compra, pois, segundos os entrevistados, o hábito leva ao conhecimento do espaço físico, da distribuição dos produtos e dos horários com menos congestionamentos da loja. Os clientes de supermercados mais automatizados valorizam, como primeira opção, o atendimento rápido e funcionários educados. Por outro lado, os clientes de supermercados menos automatizados valorizam mais o preço e a proximidade do estabelecimento. Em ambos os casos, preços e longas filas são apontados como os fatores mais negativos. [ABE 3]

No presente trabalho, admite-se que não se saiba como a demanda de consumidores do supermercado varia em função do tempo médio de espera em fila, e também supomos que não dispomos de informações *on-line* sobre o número de consumidores no estabelecimento ao longo do tempo. Os caixas são separados em caixas normais e caixas rápidos, estes últimos atendem exclusivamente consumidores com até 10 itens. Os caixas normais atendem em geral consumidores com mais de 10 itens. Para cada tipo de caixa, admitimos que os consumidores

chegam durante um dado período de tempo conforme um processo de Poisson². Os tempos de serviço, tanto dos caixas normais quanto dos rápidos, são supostos exponencialmente distribuídos. [ABE 4]

Os consumidores ao se depararem com o sistema existente podem desistir da compra e buscar outro estabelecimento, se as filas dos caixas estiverem grandes ou demoradas. Pode-se considerar também, que os consumidores escolhem sempre o caixa com menor fila e que mudam desse local durante o tempo de espera, dependendo do tamanho da fila nos outros caixas.

Segundo Bruce Schechter³, em 1974, a estréia dos *scanners* de código de barras revolucionou os supermercados. Eles não só fizeram com que as filas dos caixas diminuíssem como também melhorou consideravelmente a precisão do *check out*⁴, consistindo-se em uma fonte de informações de valor inestimável sobre *markentig* e estoques. [IBM 5]

Contudo, alguns problemas surgiram. Os códigos de barras são grandes demais para serem colados em frutas e verduras. Isso faz com os supermercados tenham bastante trabalho para identificar cada unidade com uma etiqueta ou fita torcida, ambas difíceis de remover. Algumas verduras de folhas resistem até mesmo a essas medidas, o que faz com que os funcionários dos caixas sejam treinados para distinguir a alface americana da alface comum, por exemplo. E cada vez que eles se confundem, precisam vasculhar um caderno com fotos 3x4 de frutas e verduras, enquanto os clientes esperam impacientes. [IBM 6]

2.1 Encaminhamento de Solução

Esta monografia tem o propósito de apresentar uma solução alternativa aos problemas mencionados, utilizando um protótipo para a leitura do código de barras de mercadorias, na medida em que forem sendo colocadas no carrinho de compras. A proposta para a contabilização dos produtos escolhidos na compra é a utilização

² Processo estocástico contínuo qualquer, podendo ser contínua ou descontínua.

³ Matéria disponível em http://www-03.ibm.com/e-business/br/innovation/cases/veggie_vison.shtml

⁴ Procedimento de checagem de itens que foram previamente estabelecidos por outrem.

de uma caneta ótica, que adicione na memória o produto e seu preço na sequência em que são escolhidos pelo cliente.

A idéia básica é que o próprio cliente contabilize suas compras na caneta: faz-se a leitura do código de barras, sendo o valor, dado da leitura, somado ao valor parcial já armazenado na memória. Ao mesmo tempo em que mostra o total registrado no display LCD da caneta, apresenta também a identificação e o preço do novo produto adquirido. Estes dados permanecem armazenados no PIC, sendo mantido no display até a próxima leitura. Sempre que um novo produto for lido, soma-se ao valor mostrado anteriormente no display tendo-se sempre o valor total das leituras dos produtos. O cliente também poderá navegar na lista de produtos lidos pela caneta, por meio dos botões de comando indicados pelos símbolos de ↑ e ↓. Dessa forma poderá visualizar no display qualquer produto já lido e o seu valor.

O cliente recebe sua caneta ótica ao entrar no estabelecimento, e com esta em mãos efetuará a leitura do código de barras dos produtos escolhidos. Se o cliente desejar retirar de suas compras um item já contabilizado, deverá refazer a leitura do código de barras desse artigo. Essa entrada fará a subtração no total geral, do produto relido, sem alterar a soma dos demais itens a serem aceitos (comprados). Assim, poderá controlar os valores de sua compra no ato da escolha das mercadorias.

Ao término das compras o cliente terá no display o valor total a ser pago no caixa. Este valor está armazenado no microcontrolador da caneta, e poderá ser extraído a qualquer momento. Para isto, o cliente deverá plugar um pen drive no dispositivo USB da caneta ótica, e pressionar o botão de extração de dados, situado ao lado do display na caneta. Os dados são extraídos através da porta USB e salvos em arquivo criptografado, em modo texto (arquivo .txt) no pen drive conectada na porta USB da caneta. Então, o caixa recebe o pen drive, insere em seu equipamento padrão e extrai o arquivo, com o valor total das compras do cliente.

Para a proteção anti-fraude do estabelecimento comercial, deverão ser instalados portais magnetrônicos nos caixas, que acusarão com um alarme sonoro algum produto que não esteja desmagnetizado pela caneta. Dessa forma, nenhum cliente poderá passar com o carrinho de compras no caixa (e pelo portal magnetrônico) sem antes efetuar a leitura do código de barras do produto.

2.2 Conseqüências do Problema

A cada tempo que passa, vem se tornando um transtorno para grande parte da população que, conforme já dito antes, não pode deixar de efetuar suas compras semanalmente e/ou mensalmente. E são justamente as filas que fazem do ato de fazer compras num verdadeiro transtorno na vida de um consumidor. Segundo o jornal Folha do Estado, de publicação em março de 2007, há muito os consumidores estão reclamando do tempo de espera nos caixas de supermercados. Em dias de grande movimento eles chegam a aguardar cerca de uma hora para serem atendidos. [FOL 7]

Na mesma matéria publicada em 16/03/2007, no Jornal Folha do Estado, uma dona casa reclama que a situação está ficando insuportável. Segundo ela, “além de passar muito tempo na fila ainda temos que empacotar”. Há também o comentário de outro consumidor, que diz que levou cerca de vinte minutos na fila e acha que saiu no lucro. Segundo ele, geralmente na hora do almoço perde cerca de 40 minutos na fila, então diz que vinte minutos acabam sendo razoável para ele, pois sempre que vai ao supermercado se programa para este atraso. [FOL 8]

Vários consumidores reclamam que as filas nos supermercados estão piores que as dos bancos, pois nunca conseguem ser atendidos em menos de trinta minutos. Segundo ainda a matéria divulgada no jornal Folha do Estado, o problema está no número reduzido de atendentes, que os caixas são muitos e os atendentes são poucos. A matéria ainda destaca a falta de embaladores, que torna a situação ainda pior, trazendo conseqüências para a vida dos clientes, atrasando seus compromissos. [FOL 9]

2.3 Fatores Identificados

Um estudo de caso foi realizado por Reinaldo Morabito Neto e Flavio Costa Ribeiro de Lima, da Universidade Federal de São Carlos e Universidade Estadual de Campinas respectivamente, em 2000, num supermercado da rede Jaú-Serve,

localizado em São Carlos, SP. As lojas estão sendo modernizadas, com *check-outs* de leitura ótica, consultores de preço eletrônicos, etc., e a credibilidade e fidelidade dos seus consumidores é avaliada por meio de pesquisas de opinião. [ABE 10]

As amostras relatadas a seguir foram coletadas num dia de sábado, das 14:00 às 18:00 horas. Durante este período, o sistema operou com cinco caixas normais e dois caixas rápidos. Para cada caixa, foram coletados os instantes de chegadas de consumidores na fila, de eventuais desistências durante a espera, e de início e término de serviços (ordem de centenas de observações). A tabela 1 apresenta os intervalos médios entre chegadas e os tempos médios de serviço dos consumidores para cada caixa. A média geral dos intervalos médios entre chegadas nos caixas normais é de sete minutos, e nos caixas rápidos é de 3 minutos e meio. A média geral dos tempos médios de serviço nos caixas normais é de quatro minutos e meio. A última coluna da tabela apresenta os tempos médios de espera em fila obtidos pela amostra (e não pelos modelos). [ABE 11]

Tabela 1 – Intervalos médios entre chegadas e tempos médios de serviço dos consumidores nos caixas.

Caixa	Intervalo médio Entre chegadas (min)	Tempo médio de serviço (min)	Tempo médio de espera em fila (min)
1	8,6	5,6	3,2
2	6,5	4,3	1,9
3	5,8	4,5	1,6
4	7,1	4,8	2,5
5	6,6	4,2	1,6
Média	7,0	4,6	2,16
Desvio Padrão	1,06	0,56	
6	3,3	1,6	0,9
7	3,7	1,5	1
Média	3,5	1,55	0,95
Desvio Padrão	0,29	0,07	

Fonte: disponível em <http://www.abepro.org.br>

Quase oito anos após este estudo, as filas dos caixas apresentam um tempo maior de espera e de tamanho, pois aumentou o número de pessoas que frequentam os estabelecimentos. Porém, o sistema do caixa praticamente continuou o mesmo.

Capítulo 3. REFERENCIAL TEÓRICO E BASES METODOLÓGICAS

Neste capítulo são apresentados os microcontroladores utilizados, as diferenças entre eles e suas tarefas, assim como o display LCD e a caneta ótica. Esta, uma caneta própria para leitura de código de barras existente no mercado, na qual serão acoplados chips necessários para que ela desenvolva a função de ler e mostrar o valor no display LCD. Neste capítulo são apresentadas ainda as funções dos componentes, suas utilidades e porque são necessários para este projeto.

Dos componentes acoplados, os microcontroladores são responsáveis pela parte de execução do projeto. Como citado anteriormente, eles são responsáveis pelas transferências e armazenamentos de dados (ou bits de dados) da caneta. Na verdade, os microcontroladores são responsáveis pelo controle e funções de outros componentes, como visor LCD, LED's, teclado, componente emissor de som, ou seja, todos os outros componentes eletrônicos que desempenham alguma função para o sistema de leitura ótica e interpretação.

Os microcontroladores estão escondidos dentro de inúmeros produtos atualmente. Se um forno de microondas tem um LED ou visor LCD e teclado, ele contém um microcontrolador. Todos os automóveis modernos contêm ao menos um microcontrolador (e podem ter até 6 ou 7): O motor é controlado por um microcontrolador, bem como os freios antitravamento, o controle de velocidade de viagem e assim por diante. A maioria dos aparelhos com controle remoto contém um microcontrolador: TV's, VCR's, e sistemas de som de alta fidelidade entram nesta categoria. Além disso, têm microcontroladores: câmeras SLR e câmeras digitais, telefones celulares, filmadoras, impressoras a laser, telefones (as que possuem identificador de chamadas, memória para vinte números, etc.), pagers, além de refrigeradores, lava-louças, lavadoras de roupas e secadoras (os que possuem visores e teclados). Basicamente, qualquer produto ou dispositivo que interaja com o usuário possui um microcontrolador interno. (RICARDO; PAULO, 2008) [JOR 12]

Há cerca de uma dúzia de pinos de I/O no PIC que podem ser configurados como entradas ou saídas. Quando configurados como saída, eles fornecem corrente suficiente para controlar um LED diretamente. Um conjunto de pinos de I/O é usado para programar a EEPROM serialmente com um programador externo. Esta memória pode ser reprogramada milhares de vezes. Existem muitos modelos de programadores disponíveis no mercado ou para montar em revistas especializadas e na Internet. (PEROTTO; MARCELO, 2005) [MAR 13]

O assunto sobre os microcontroladores será mais detalhado no tópico 3.1.1 deste trabalho, onde se têm explicações relevantes consideradas pela utilização de microcontrolador na solução proposta.

3.1 Dispositivos

O projeto valeu-se de uso de alguns dispositivos eletrônicos, para que fossem possíveis as novas funções da caneta ótica. Dispositivos eletrônicos fazem parte dos aparelhos eletrônicos modernos e atuam institivamente para a função do equipamento. Tais dispositivos são responsáveis por acréscimos de funções em alguns equipamentos já existentes, e o mesmo acontece para com este projeto.

A idéia básica é acoplar dispositivos eletrônicos a um equipamento já existente, para que este seja capaz de efetuar novas funções. Neste projeto em questão, o acoplamento de dispositivos como microconrolador, display LCD e driver para pen drive farão com que um scanner ótico (caneta ótica) tenha agora funções de armazenamento de dados, contabilização de dados decodificados pelo leitor ótico, visualização de dados no próprio equipamento (através do display LCD) e extração desses dados através de porta USB. A caneta ótica utilizada existente no mercado, efetua leitura de código de barras e envia os dados da leitura através de uma saída serial. Neste projeto a caneta é capaz de armazenar os dados lidos, mostrá-los na tela e efetuar a contabilização (soma) dos valores lidos até chegar a um valor total, que nada mais é do que a última leitura. A extração desses dados também é feita por componentes eletrônicos acoplados na caneta ótica.

Para a demonstração da técnica e metodologia utilizada, os tópicos a seguir explanam sobre os dispositivos utilizados para promover a nova função de uma caneta ótica convencional em uma caneta ótica ou scanner ótico que registra e contabiliza produtos automaticamente.

3.1.1 Microcontroladores

Os microcontroladores (MCU's) são dispositivos mais simples, com memória RAM (armazenamento de dados e variáveis), memória ROM (armazenamento do programa), oscilador interno de clock, I/O interno, entre outros. Por essa razão são chamados computadores em um único chip.

Os microcontroladores atuam como se fossem computadores de propósitos específicos. São “embutidos” no interior de algum outro dispositivo (geralmente um produto comercializado), para que possam controlar as ações ou funções do produto. Também podem ser chamados de controlador embutido. São dedicados e executam um programa específico. O programa é armazenado na memória ROM e geralmente não muda. São dispositivos de baixa potência, pois são alimentados por bateria e podem consumir 50 miliwatts.

Os microcontroladores são diferentes dos processadores, pois integram em um único CHIP: o processador, uma memória RAM (dados), uma memória ROM (programa) e outros recursos.

Os microcontroladores possuem um dispositivo dedicado de entrada e geralmente possuem um pequeno LED ou visor LCD de saída, e também obtém a entrada do dispositivo que está controlando, enviando sinais a diferentes componentes desse dispositivo. Geralmente são pequenos e baratos. Os componentes são escolhidos para minimizar o tamanho e serem os mais econômicos possíveis. São geralmente feitos para serem robustos. Um chip microcontrolador simples pode conter mil bytes (1.000 bytes) de ROM e vinte bytes (20 bytes) de RAM, juntamente com 8 pinos de E/S (Entrada e Saída). Em grandes quantidades, o custo desses chips pode ser o menor possível (de dezenas de centavos a alguns reais). [JOR 14]

Esses microcontroladores são encontrados em uma série de soluções eletro/eletrônicas, sendo de vários tipos e modelos, cada qual com funções e utilidades a que a solução se refere, mas sempre utilizado para controle de entrada e saída de bits e dados de demais componentes, fazendo com que seja o coração de um equipamento ou solução eletrônica.

Na figura 3.1 está ilustrado um microcontrolador contido em aparelhos domésticos.

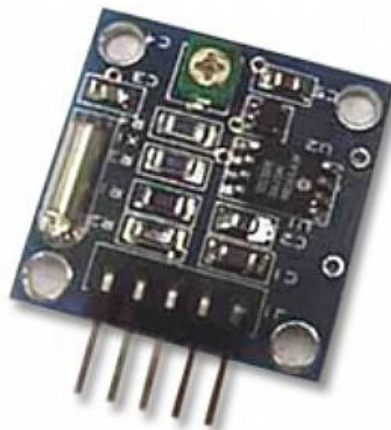


Figura 3.1 – Imagem de um Microcontrolador de aparelhos domésticos

Na figura 3.2 é apresentado um exemplo de microcontrolador também a ser acoplado na caneta ótica, responsável por múltiplas tarefas, como transporte de dados, armazenamento de dados, envio de dados por meio de bits elétricos para outros componentes eletrônicos, controle de bits e dados.



Figura 3.2 – PIC 16F84.

Outro exemplo de microcontrolador é ilustrado na figura 3.3. Tanto o microcontrolador ilustrado na figura 3.1 quanto o microcontrolador ilustrado na figura 3.3 podem ser utilizados nesse tipo de equipamento.



Figura 3.3 – PIC BASIC Step 1

Também, é utilizado um driver para pen drive, que serve como porta de extração de dados. Neste caso, uma porta USB, por onde é inserido um pen drive na caneta. Os dados finais são exportados por este driver. Este tipo de opção pode facilitar o manuseio da caneta, uma vez que o estabelecimento pode adotar um local único para devolução das canetas óticas e o consumidor pode carregar consigo apenas o pen drive para entregar ao caixa. Caso o estabelecimento não adote esta opção, o próprio atendente do caixa pode extrair os dados da caneta para o pen drive, com o consumidor devolvendo a caneta diretamente ao caixa. Neste caso, o atendente extrai o valor total da compra para o pen drive da caneta, e conecta o pen drive em seu PC para visualização e conferimento do valor total da compra do cliente. Na figura 3.4 é ilustrado um driver para pen drive que é acoplado na caneta para efetuar as funções comentadas.



Figura 3.4 – PenBS – Driver para Pen Drive

Atualmente, o que ocorre com o tipo de leitor de código de barras usado é que só pode ser manuseado pelo operador do caixa, pois permanece sempre conectado ao terminal do caixa por um cabo serial. Dessa forma, a leitura dos códigos de barras dos produtos comprados só pode ser efetuada pelo operador.

3.1.2 Display LCD

Existem dois tipos diferentes de display: os displays LED (de sete segmentos ou matriz de pontos), e os módulos LCD, tornando-os adequados para cada aplicação definida.

No caso, o Display LCD, com vantagem do baixo consumo de energia, permite uma grande concentração de informações em um pequeno espaço físico, de forma relativamente simples e barato. O módulo display LCD é constituído basicamente de um display de cristal líquido.

Os módulos LCD são interfaces de saída muito úteis em sistemas microprocessados. Esses módulos podem ser gráficos e caracteres (comuns). Os módulos LCD gráficos são encontrados com resoluções 132x32; 128x64; 240x64 e 240x128 dots pixel, e geralmente estão disponíveis para 20 pinos para conexão. Por

serem mais caros e complexos, podem apresentar qualquer tipo de informação na tela, como por exemplo, fotos, além dos próprios gráficos. Os LCD comuns (tipo caractere) podem apresentar em forma de letras, números e símbolos. Sua tela é dividida em linhas e colunas e em cada posição é armazenado um caractere, conforme configurações previstas na Tabela 2. [MIC 15]

Tabela 2 – Módulos LCD disponíveis

Número de Colunas	Número de Linhas	Quantidade de pinos
8	2	14
12	2	14/15
16	1	14/16
16	2	14/16
16	4	14/16
20	1	14/16
20	2	14/16
20	4	14/16
24	2	14/16
24	4	14/16
40	2	16
40	4	16

Fonte: artigo de L., Ilton; Fleury, Cláudio – outubro 2006

Nesta apresentação, foi trabalhado com os módulos de caractere, baseado no chip controlador HD 44780. Esse chip é considerado padrão quando se trata de módulos de display LCD. Esses controladores (HD 4478) permitem uma interface simples com sistemas microprocessados ou microcontroladores. Nesses casos, com largura de barramento de dados entre 4 ou 8 bites selecionáveis, com mais três linhas de sinalização adicionais: ENABLE, RS e R/W. [MIC 16]

Para se usar uma comunicação no modo de 4 bites, utiliza-se as quatro linhas mais significativas de dados (D7 a D4), dividindo então o byte em dois nibbles, que são transferidos, iniciando-se sempre pelo mais significativo, seguido pelo menos significativo.

Com um total de 208 caracteres 5x8 mais 32 caracteres 5x10, esses controladores dispõem ainda de 80 bytes de memória RAM (DDRAM – Data Display RAM de dados do display), 64 bytes de RAM para o gerador de caracteres do usuário (CGRAM – Character Generator RAM do gerador de caracteres) e 9920 bites de memória ROM (do gerador de caracteres – CGROM).

Na figura 3.5 é ilustrado um exemplo de um display LCD, que é acoplado à caneta ótica para ter a função de mostrar o valor contabilizado do produto. Com isso, a cada leitura, o consumidor irá acompanhando o valor de sua compra pelo display LCD.



Figura 3.5 – Display LCD. Fonte: <http://www.tato.ind.br>

3.1.3 A Tecnologia Microcontrolada

As técnicas metodológicas dos microcontroladores para a programação em linguagem C, abrange a otimização do PIC 16F648A, interpretações, interfaceamentos com o display LCD, comunicação, entre outros.

A utilização da tecnologia em linguagem C é muito eficiente para a programação de microcontroladores, sendo uma escolha natural para o desenvolvimento de software.

Antes, porém, é preciso entender sobre programação e dispositivos programáveis. Programar é fornecer uma sequência determinada de comandos ou instruções para ser executada por uma pessoa ou máquina. Podemos dizer agora

que os dispositivos programáveis são as instruções sequenciais do programa, determinados para ser executado pela máquina.

A programação de um módulo LCD de caracteres utilizando o HD 44780 ou equivalente, obedece a um conjunto de comando.

Para a programação de caracteres no CGRAM utiliza-se primeiro o registrador AC, que é carregado com o endereço inicial de CGRAM (comando 0x40). Um procedimento simples de ser realizado. Em seguida, escreve-se o dado correspondente à primeira linha de caractere a ser programado.

Na programação é preciso que um bit 1 signifique pixel ligado e bit 0 pixel desligado. E que os três bits mais significativos (7,6 e 5) devem ser deixados em zero. Como também, dependendo do tamanho da matriz, são enviados os dados das próximas linhas de caracteres correspondentes, até a oitava ou décima. É preciso lembrar que a última linha é deixada em branco (0x00). Nesta posição, o cursor é ativado pelo hardware. Após efetuar a programação da CGRAM é preciso lembrar também de alterar o registrador AC, para a região da memória DDRAM. A tabela 3 exemplifica a construção de uma seta para cima, na primeira posição da CGRAM.

Tabela 3 – Construção de uma seta para cima

RS	R/W	DADO	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0x04	0	0	0	0	0	1	0	0
0	0	0x0E	0	0	0	0	1	1	1	0
0	0	0x15	0	0	0	1	0	1	0	1
0	0	0x04	0	0	0	0	0	1	0	0
0	0	0x04	0	0	0	0	0	1	0	0
0	0	0x04	0	0	0	0	0	1	0	0
0	0	0x04	0	0	0	0	0	1	0	0
0	0	0x00	0	0	0	0	0	0	0	0

Fonte: Pereira, Fábio. Microcontroladores PIC Programação em C. 2º Edição. Editora Érica, 2003, página 314

3.1.3.1 O Microcontrolador PIC 16F648A

Este modelo de MCU foi escolhido para o projeto caneta ótica que contabiliza os produtos automaticamente porque ele é versátil, compacto, rápido, poderoso e de fácil aprendizagem. As funções necessárias para o desenvolvimento e produção da caneta ótica são compatíveis com o uso do PIC 16F648A, e é utilizado no projeto tanto na parte de testes quanto na parte definitiva. Suas principais características são: 16 pinos I/O; frequência de até 20Mhz; oscilador interno de 4Mhz/37Khz; 4kx14 bits de memória FLASH de programa; 256x8 bits de memória EEPROM interna; módulo TIMER⁵ 0 (timer /contador de 8 bits); módulo TIMER 1 (timer /contador de 16 bits); módulo TIMER 2 (timer de 8 bits); módulo CCP; módulo de referência de tensão; dois comparadores analógicos com referência interna programável de tensão; um canal de comunicação serial; um watchdog timer⁶; um canal de comunicação USB; capacidade de corrente de 25 mA por pino de I/O; 35 instruções; dez fontes de interrupção independentes; modo de funcionamento sleep; tensão de operação entre 3v a 5v. [PIC 17]

Na figura 3.6 está ilustrada a imagem de um microcontrolador PIC 16F648A.

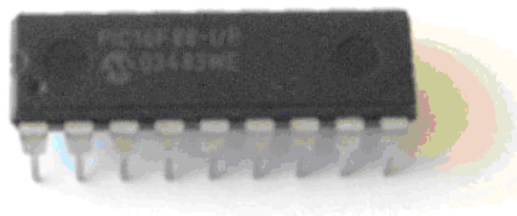


Figura 3.6 – Microcontrolador 16F648A

Na figura 3.7 é ilustrada a pinagem de um microcontrolador PIC 16F628A. O microcontrolador PIC 16F648A possui as mesmas características do PIC 16F628A,

⁵ Contador de tempo em sistemas.

⁶ Watchdog timer é um dispositivo eletrônico temporizador que dispara um reset ao sistema ao programa principal devido a alguma condição de erro do programa.

inclusive a pinagem, sendo a única diferença entre os dois modelos o tamanho da memória, pois conforme explicado anteriormente, o microcontrolador 16F648A possui o dobro da memória do microcontrolador 16F628A.

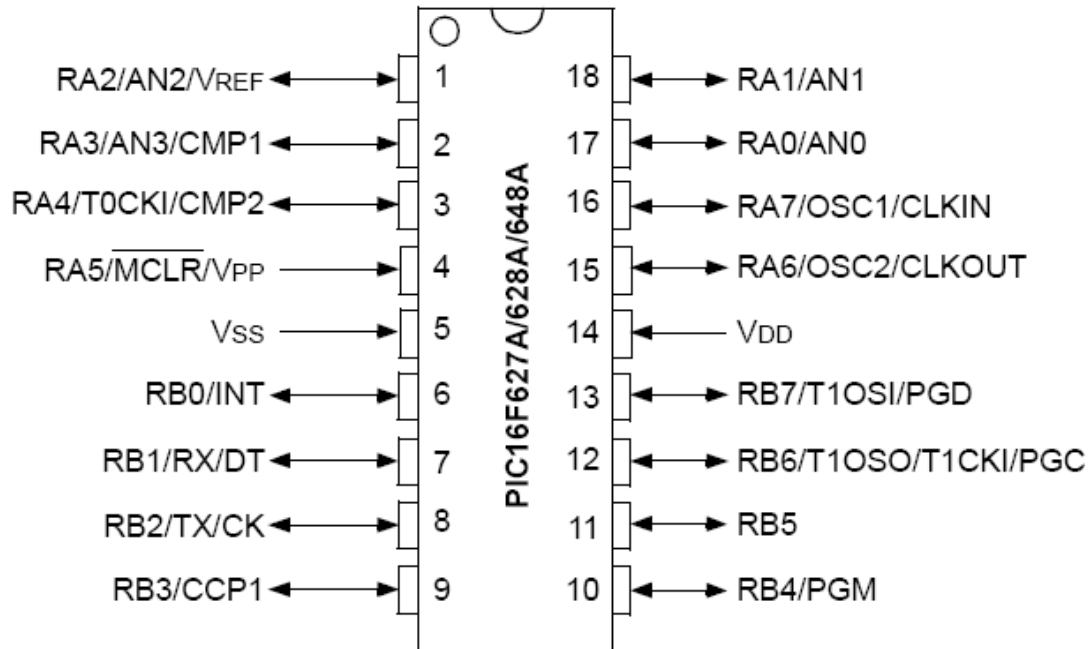


Figura 3.7 – Pinagem do PIC 16F628A

Toda CPU executa várias tarefas de uma só vez, e ao mesmo tempo. Essas tarefas são denominadas de instruções e são executadas a partir de um sinal de clock gerado por um circuito oscilador. Quando a sequência de valores volta a se repetir, é chamada de ciclo de clock. Cada instrução que o PIC 16F648A executa consome quatro ciclos de clock, denominado de ciclo de instrução. Todas as instruções, com exceção daquelas que provocam desvio no programa, como CALL e GOTO, são executadas em um ciclo de instrução. As instruções que provocam desvio no programa são executadas em dois ciclos de instrução. Isso torna o sistema muito mais rápido e só é possível porque o barramento de dados é separado do barramento de instruções de programa, de acordo com a arquitetura Harvard⁷, adota nos MCUs da Microchip⁸. O circuito oscilador do PIC 16F648A pode

⁷ Modelo de arquitetura para hardware.

funcionar em oito diferentes modos de operação. A escolha do modo de operação é feita por meio de três bits de configuração. Essa configuração é feita no programa-fonte por meio de uma sintaxe própria, ou na hora de efetuar a gravação do programa no chip. A figura 3.8 mostra a janela de configuração do modo de operação do oscilador do PIC 16F648A, no programa MPLAB IDE versão 6.40. [PIC 18]

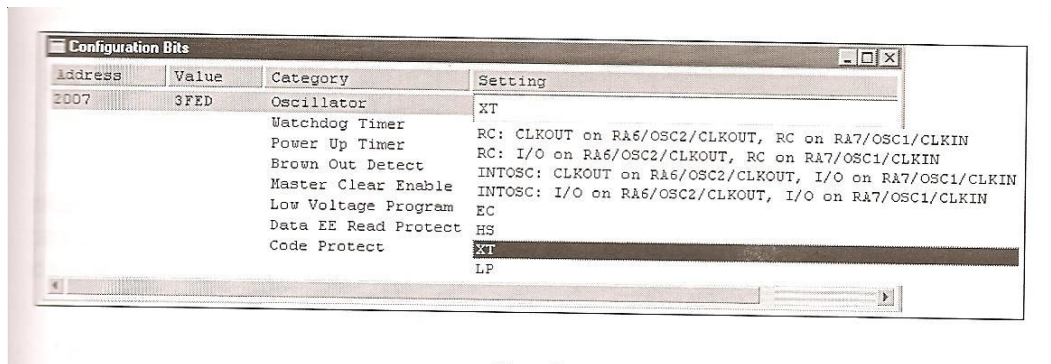


Figura 3.8 – Modo de operação do oscilador do PIC 16F648A

3.2 Leitor de Código de Barras (Barcode Scanner)

O leitor de código de barras é um equipamento ótico que faz a leitura das barras impressas sobre uma superfície plana. Um feixe de laser faz a varredura da superfície e converte as informações para uma codificação binária, decodificando a informação contida no código de barras.

Na figura 3.9 é ilustrado um leitor de código de barras portátil, de modelo convencional. Este tipo de leitor de código de barras é encontrado em qualquer estabelecimento comercial atual, e pode ser comprado por comerciantes que desejam tê-lo em seu estabelecimento. É encontrado para venda em lojas especializadas em vendas de equipamentos eletrônicos.

⁸ Marca de revendedor de microcontroladores.



Figura 3.9 – Leitor de código de barras portátil

3.3 Código de Barras – Conceitos Básicos

Código de barras nada mais é do que uma representação gráfica de dados, que podem ser numéricos ou alfanuméricos. A leitura desses dados é realizada pela decodificação por um aparelho chamado “scanner”, que emite um raio vermelho que percorre todas as barras. Ao percorrer as barras, a luz é absorvida onde a barra for escura, e refletida onde a barra for clara (representa os espaços). Com a reflexão da luz clara, a luz é refletida novamente para o “scanner”, reconhecendo assim os dados que ali estão representados.

Para compreender como as informações são codificadas em código de barra é fundamental que se tenham os seguintes conceitos abaixo (conceitos retirados do site http://www.macoratti.net/cod_bar.htm, acessado em 12/10/2008):

Barra – Consiste na parte escura do código (normalmente preta). Ela absorve a luz e codifica um em cada módulo de barra.

Espaço – Consiste na parte clara do código (geralmente o fundo que o código é impresso), ele reflete a luz e cada módulo é codificado como zero.

Caractere – Caracteriza cada número ou letra codificados com barra e espaço. Cada caractere pode ser modificado por tantos “1” ou “0” quantos forem os módulos contidos na sua codificação.

Caractere inicial e final – Indicam ao leitor do código, respectivamente o início e o fim do código. Estes caracteres podem ser representados por uma letra, um número ou outro símbolo dependendo do padrão do código em questão.

Separadores – Os separadores servem para indicar as extremidades do código e indicar ao leitor o sentido que o código está sendo lido. Estes separadores permitem também que o código seja lido nos dois sentidos.

Zonas mudas – Considerada de suma importância para o reconhecimento do código pelo leitor ótico, uma vez que a sua exclusão no momento da leitura poderá ser interpretada pelo código de barra com leitura inexistente ou nula. São espaços existentes antes do caractere inicial e depois do caractere final, em forma de margens. Também conhecida com “quites zonas”.

Sinais de enquadramento – Todos os elementos do código devem estar contidos numa área retangular, no espaço delimitado somente para esse código de barra.

Densidade do código de barra - Significa a caracterização da relação entre a quantidade de módulos ou caracteres e os espaços, ocupados por eles, uma vez impressos no código de barra.

Módulo – Caracterizado pela unidade mínima e básica que compõe o código de barra. Seu tamanho é definido diretamente pela densidade do código. É considerado o elemento mais estreito do código, podendo ser um espaço ou uma barra. Todos os elementos que compõem o código de barra (os separadores, as zonas mudas, os caracteres especiais) são múltiplos do módulo quanto a largura.

FLAG – Localizado no início do código, ele é caracterizado, no sistema EAN (adotado como padrão pela ABAC⁹), para indicar o país de origem. Como também no UPC, indica o tipo de produto no início do código, caracterizado pela localização.

Dígito verificador – Constituído por um elemento incluído no código, é utilizado para detectar erros durante a varredura. Serve para evitar adulterações ou

⁹ Associação Brasileira de Automação Comercial

a leitura errônea do código. Esse dígito verificador (elemento) é calculado a partir de um algoritmo, que localiza os demais números do código.

Como os dados são codificados em barras e como os códigos de barras são lidos e interpretados.

Convencionalmente, a identificação do dígito 0 (zero) se dá pela sequência de cinco barras: duas estreitas, duas largas e uma estreita, espaçadas pelos espaços em branco. Como exemplo, o dígito 0 codificado seria apresentado da seguinte maneira:

<i>Dígito</i>	<i>Código</i>
0	

Figura 3.10 – Codificação do dígito 0.

Os dígitos de 0 - 9 ficaria :

<i>Dígito</i>	<i>Código</i>
0	
1	
2	
3	
4	

Figura 3.11 – Codificação dos dígitos 0 a 4.

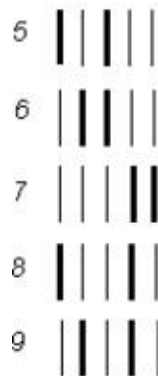


Figura 3.12 – Codificação dos dígitos 5 a 9.

O número 1998 ficaria:

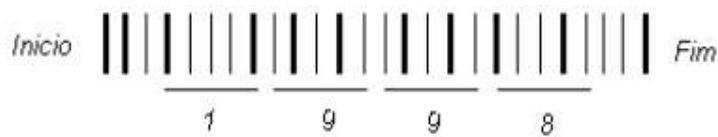


Figura 3.13 – Codificação do numero 1998.

Assim, pode-se fazer a leitura no código de barras. Primeiramente, há necessidade de se entender os conceitos sobre leitura e decodificação.

Uma caneta ótica em funcionamento transforma os dados codificados em barras em sinais elétricos, de acordo com a duração proporcional dessas barras, através de um dispositivo de leitura ótica. Isso porque esse dispositivo citado tem em sua ponta um emissor de luz, normalmente vermelho, e, um sensor para receber a reflexão dessa luz (ou não).

Como exemplo, se a caneta ótica for direcionada para uma superfície branca, o sensor vai captar uma reflexão gerada pela superfície, gerando um sinal elétrico, correspondendo a essa superfície branca. Por outro lado, se a caneta for colocada em cima da superfície preta, a luz emitida pelo emissor será absorvida. Com isso, o sensor deixará de receber reflexo, gerando um sinal correspondente a cor preta. Desse modo, se a caneta ótica for posicionada sobre uma etiqueta de código de barra, numa velocidade constante, notadamente, essa caneta irá gerar uma sequência de sinais elétricos, representadas pelas barras pretas largas e

estreitas e espaços em branco. Por essa razão, o código de barra é interpretado pelo leitor, indicado através da luz, refletida pelos módulos que compõem os espaços, ou pela sua ausência nas barras.

Para essa interpretação de leitura, utiliza-se um conversor analógico/digital que transforma esses sinais elétricos analógicos produzidos pela luz recebida pelo sensor fotoelétrico, em sinal digital (sucessão de 0 e 1 em forma de pulso). Sabendo-se que na ausência da luz refletida é gerado outro sinal digital, caracterizando a barra. Dessa maneira, cada caractere do código é interpretado como um número binário, onde cada módulo reproduz um dígito 0 (espaço no código) ou 1 (absorção da luz).

Para a decodificação, de acordo com o padrão do código que foi utilizado, os sinais elétricos de um dispositivo leitor de código (caneta ótica) são transformados em um correspondente caractere. Como exemplo, o dígito 0 (zero) como o lido pelo leitor ótico, por ser o de mais fácil compreensão, para entender os demais dígitos.

Conforme citado anteriormente, o dígito 0 é caracterizado pelo código contendo duas barras estreitas, duas largas e uma estreita e ainda, utilizando-se a tabela de codificação do código 25 (utilizado no Brasil), ao passar pelo conversor, ele indicará o caractere 0.

Os códigos de barras mais utilizados no Brasil são: 25 intercalado; 39; 39 com dígito de verificação, EAN-8 e EAN-13. Cada um com as suas especificidades de aplicações. No caso, será abordado somente o EAN-13, sendo a codificação mais utilizada nos produtos, pelos supermercados, objeto da monografia. O EAN-13 codifica o país de origem, o fabricante e o código do produto. Desenvolvido pela Internacional Article Numbering Association (EAN) e sua utilização é regida pela GSL, cujas normas são internacionais. A simbologia EAN-13 também é conhecida pelos termos GS1-13, EAN/UCC-13, GTIN-13, European Article Number 13 (termos originais em inglês).

Existe também o código do tipo GS1-128¹⁰ (anteriormente intitulado de UCC/EAN-128), que é um código de barras linear alfanumérico que representa identificadores de aplicação e permite a inserção de uma maior quantidade de informação e dados adicionais, assim como números de série, datas de validade, ou

¹⁰ Referência ao tipo de código de barras.

medidas e também o número de lote de produção. É um tipo de simbologia de identificação de produtos com maior necessidade de serem rastreados e que é especialmente de utilização logística e para produtos hospitalares. [GS1 19]

Na figura 3.14 é ilustrado o tipo de código de barras GS1-128. Por este exemplo, pode-se notar a diferença da disposição alfanumérica do código de barras para o tipo EAN-13 (mais utilizado para produtos de prateleiras). Outra diferença fundamental do código GS1-128 é que nele há especificado o prazo de validade do produto, além do número do lote e o número de série do produto.



Figura 3.14 – Exemplo de um GS1-128

A título de informação, na tabela 4 estão exemplificados os identificadores de aplicação (IA – prefixo numérico utilizado para assinalar e identificar o tipo de informação e o formato em que se encontra o código de barras), a descrição, número de dígitos e o formato do código de barras do tipo GS1-128.

A melhor maneira de definir o que realmente são os Identificadores de Aplicação é através da citação de PAIXÃO (2006, p.32), que diz que estes são "pequenos segmentos de informação normalizada e aceites globalmente, que se vão intercalar com os dados propriamente ditos". A maior parte da informação necessária para uma eficiente rastreabilidade dos produtos encontra-se com um número de Identificador de Aplicação atribuído. [PAI 20]

Tabela 4 – Tabela de Identificadores de Aplicação – código de barras tipo GS1-128.

IA	Descrição	Número de dígitos e formato
00	<i>Serial Shipping Container Code (SSCC)</i>	18 dígitos numéricos
01	<i>Global Trade Item Number (GTIN)</i>	máx. 4 dígitos numéricos
02	Número de contentores	14 dígitos numéricos
10	Número <i>Batch</i>	de 1 a 20 dígitos alfanuméricos
11	Data de produção	6 dígitos numéricos no formato AAMMDD
13	Data de embalagem	6 dígitos numéricos no formato AAMMDD
15	Melhor antes de... (usado no controlo da qualidade)	6 dígitos numéricos no formato AAMMDD
17	Data de validade	6 dígitos numéricos no formato AAMMDD
20	Variável do produto	2 dígitos
21	Número de série	de 1 a 20 dígitos alfanuméricos
22	Quantidade HIBCC, data, <i>Batch</i> e ligação	de 1 a 29 dígitos alfanuméricos
23x	Número de lote	de 1 a 19 dígitos alfanuméricos

240	Identificação adicional do produto	de 1 a 30 dígitos alfanuméricos
250	Segundo número de série	de 1 a 30 dígitos alfanuméricos
30	Quantidade de cada	dígitos necessários
310y	Peso líquido do produto (em kg)	6 dígitos
311y	Comprimento do produto - 1ª dimensão (em metros)	6 dígitos
312y	Largura ou diâmetro do produto - 2ª dimensão (em metros)	6 dígitos
313y	Profundidade ou espessura do produto - 3ª dimensão (em metros)	6 dígitos
314y	Área do produto (em metros quadrados)	6 dígitos
315y	Volume do produto (em litros)	6 dígitos
316y	Volume do produto (em metros cúbicos)	6 dígitos
320y	Peso líquido do produto (em libras)	6 dígitos
321y	Comprimento do produto - 1ª dimensão (em polegadas)	6 dígitos
322y	Comprimento do produto - 1ª dimensão (em pés)	6 dígitos

323y	Comprimento do produto - 1ª dimensão (em jardas)	6 dígitos
324y	Largura ou diâmetro do produto - 2ª dimensão (em polegadas)	6 dígitos
325y	Largura ou diâmetro do produto - 2ª dimensão (em pés)	6 dígitos
326y	Largura ou diâmetro do produto - 2ª dimensão (em jardas)	6 dígitos
327y	Profundidade ou espessura do produto - 3ª dimensão (em polegadas)	6 dígitos
328y	Profundidade ou espessura do produto - 3ª dimensão (em pés)	6 dígitos
329y	Profundidade ou espessura do produto - 3ª dimensão (em jardas)	6 dígitos
330y	Peso bruto do contentor (em kg)	6 dígitos
331y	Comprimento do contentor - 1ª dimensão (em metros)	6 dígitos
332y	Largura ou diâmetro do contentor - 2ª dimensão (em metros)	6 dígitos
333y	Profundidade ou espessura do contentor - 3ª dimensão (em metros)	6 dígitos
334y	Área do contentor (em metros quadrados)	6 dígitos

335y	Volume bruto do contentor (em litros)	6 dígitos
336y	Volume bruto do contentor (em metros cúbicos)	6 dígitos
340y	Peso bruto do contentor (em libras)	6 dígitos
341y	Comprimento do contentor - 1ª dimensão (em polegadas)	6 dígitos
342y	Comprimento do contentor - 1ª dimensão (em pés)	6 dígitos
343y	Comprimento do contentor - 1ª dimensão (em jardas)	6 dígitos
344y	Largura ou diâmetro do contentor - 2ª dimensão (em polegadas)	6 dígitos
345y	Largura ou diâmetro do contentor - 2ª dimensão (em pés)	6 dígitos
346y	Largura ou diâmetro do contentor - 2ª dimensão (em jardas)	6 dígitos
347y	Profundidade ou espessura do contentor - 3ª dimensão (em polegadas)	6 dígitos
348y	Profundidade ou espessura do contentor - 3ª dimensão (em pés)	6 dígitos
349y	Profundidade ou espessura do contentor - 3ª dimensão (em jardas)	6 dígitos

350y	Área do produto (polegadas quadradas)	6 dígitos
351y	Área do produto (pés quadrados)	6 dígitos
364y	Volume do produto (em polegas cúbicas)	6 dígitos
365y	Volume do produto (em pés cúbicos)	6 dígitos
366y	Volume do produto (em jardas cúbicas)	6 dígitos
367y	Volume bruto do contentor (em polegas cúbicas)	6 dígitos
368y	Volume bruto do contentor (em pés cúbicos)	6 dígitos
369y	Volume bruto do contentor (em jardas cúbicas)	6 dígitos
37	Número de unidades contidas	de 1 a 8 dígitos
8001	Medidas de produtos redondos (largura, comprimento e diâmetro)	14 dígitos
8002	<i>Electronic Serial Number</i> (ESN) exclusivamente para telemóveis	de 1 a 20 dígitos alfanuméricos
8003	Número UPC/EAN e número de série de <i>Returnable Asset</i>	14 dígitos do número UPC/EAN e de 1 a 16 do número de série de <i>Returnable Asset</i>
8004	Identificação em série UPC/EAN	de 1 a 30 dígitos alfanuméricos

8005	Preço por unidade de medida	6 dígitos
90	Acordado mutuamente pelas partes envolvidas na comercialização do produto	de 1 a 30 dígitos alfa numéricos
92	Códigos internos de empresas	de 1 a 30 dígitos alfanuméricos

Fonte: GS1-128 (formerly UCC/EAN-128) Barcodes (2006)

Capítulo 4. A PROPOSTA DO MODELO (PROTÓTIPO)

Para a construção da caneta ótica que contabiliza os produtos automaticamente, foi adquirido um kit de testes de funções Acepic¹¹ (modelo 18), contendo os seguintes dispositivos: microcontroladores, leds, display analógico e display LCD, entrada serial de dados, socket para microcontroladores e botões de comando. Adquirido também um scanner ótico convencional (caneta ótica), e os microcontroladores de modelo 16F84, 16F628, 16F628A e 16F648A, para efetuar os devidos testes de funções do microcontrolador e do display LCD.

O kit de testes é geralmente utilizado para testes de software em hardware. Por uma entrada serial, pode-se gravar no microcontrolador um programa já elaborado, e depois executar esse programa. O microcontrolador atua nos leds de teste e no display LCD acoplado na placa, alternando as funções conforme executa o programa. Para os leds, podem-se fazer testes de alternância de acendimento destes, e para o display, testes de visualização de números e letras no visor. Na figura 4.1 é ilustrado o kit de teste Acepic adquirido (placa de desenvolvimento e periféricos), onde os primeiros testes de funções foram efetuados.



Figura 4.1 – Kit de testes de funções de dispositivos. Fonte: Do autor.

¹¹ Empresa fabricante de placas de desenvolvimento e placas de testes.

O kit utilizado já fornece exemplos de programas para testes, de fontes abertas, e também para implementação dos mesmos. Para isso, é utilizado o compilador MikroC, que permite otimizar os programas exemplos, aumentando assim o número de funções de testes no programa. Na figura 4.2 é exemplificado um programa simples para teste dos leds da placa de teste no compilador MikroC. O programa de teste é compilado e gravado no microcontrolador através de cabo serial, conectado na porta serial do kit. Após a transferência do programa de testes, é necessário executar o programa, pressionando o botão de execução de programas da placa de desenvolvimento. Desse modo, as funções descritas no programa são executadas pelos dispositivos da placa de desenvolvimento.

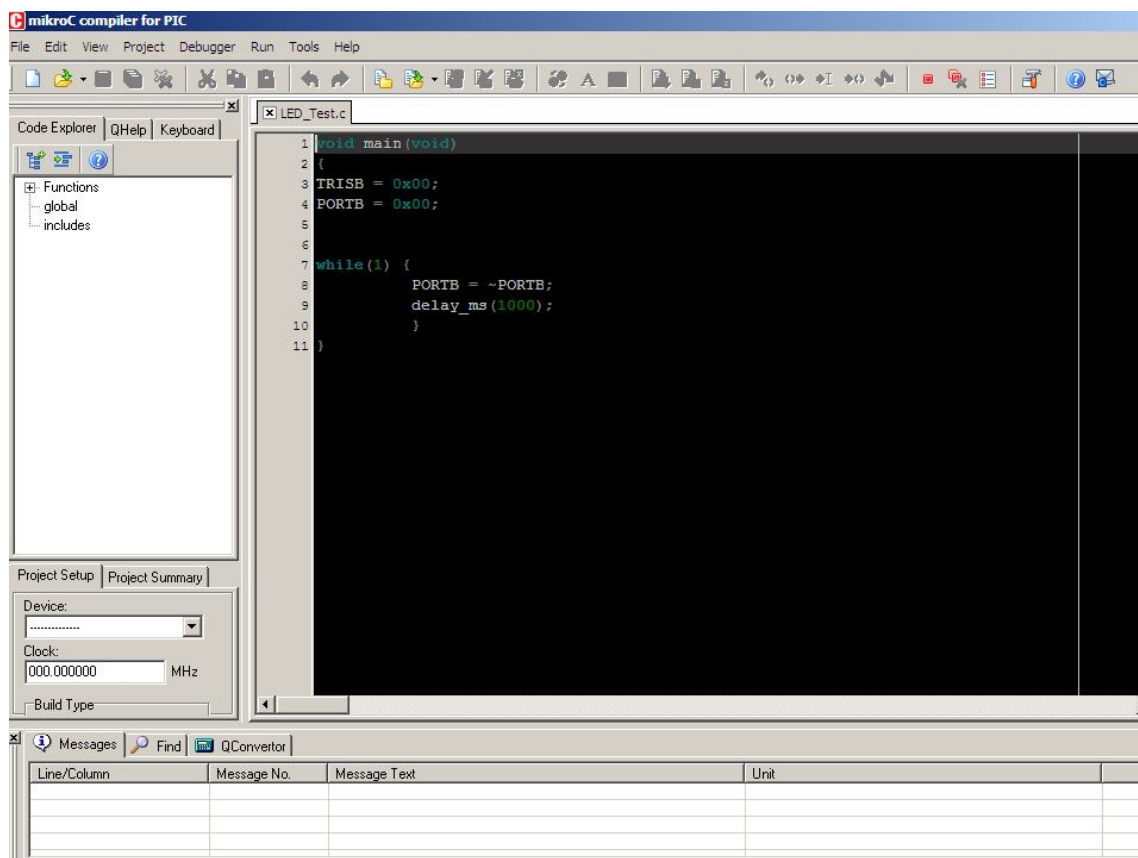


Figura 4.2 – Código de programa de teste de leds da placa de testes no Compilador Mikroc. Fonte: Do autor.

Para este projeto, foi utilizada a versão 1.05D do aplicativo IC-Prog. Na figura 4.3 é ilustrado o aplicativo em tal versão, e o código do projeto caneta ótica que registra e contabiliza os produtos automaticamente em sua versão final já em hexadecimal, pronto para ser gravado no PIC 16F648A.



A idéia para a utilização da placa de desenvolvimento foi a de testar os componentes básicos que fazem parte da placa definitiva acoplada na caneta ótica. Dessa forma, o esquema eletrônico pôde ser montado para a placa definitiva, onde os componentes necessários para tal placa permaneceram com suas funções e os demais foram descartados, além de poderem-se projetar funções necessárias para a placa definitiva. Um exemplo disso são os botões de comando projetados para a placa da caneta ótica. Pode-se projetá-los com suas funções específicas pela facilidade de tê-los já instalados na placa de desenvolvimento. Outra facilidade foi a do uso da porta serial da placa, responsável pelas conexões seriais tanto da caneta ótica convencional como pela conexão serial ao computador (comunicação e transportes de dados placa de testes-computador). O microcontrolador instalado na placa de desenvolvimento efetua o controle de bits para os outros dispositivos, controlando assim os bits das portas dos botões de comando e execução, da porta do display LCD, da porta serial, das portas dos leds, etc, sendo cada pino do microcontrolador responsável por uma determinada função/tarefa, controlando dispositivos distintos. No Anexo II deste trabalho está ilustrado o esquema elétrico da placa de desenvolvimento do projeto (páginas 86, 87 e 88), indicando as comunicações de cada pino do microcontrolador PIC com cada periférico da placa.

Para o funcionamento do projeto da caneta ótica, foi desenvolvido um programa bufferizador ¹²para o microcontrolador, responsável por receber a leitura dos códigos de barra e verificar se há um produto de mesmo código bufferizado. O programa mostrará o produto lido no display LCD assim como o seu valor no caso do produto estar bufferizado, caso contrário, enviará ao display LCD a informação “Produto Inválido”. Para estes testes de leitura de código de barras utilizando a caneta ótica convencional, o microcontrolador e o display LCD da placa de desenvolvimento, foi conectada a porta de saída serial da caneta ótica na porta de entrada serial da placa de testes, conforme ilustração da figura 4.4. Com a leitura do código de barras, a caneta ótica decodifica o código e envia os dados para o microcontrolador. Neste, o programa identifica os dados recebidos e compara com os dados bufferizados no PIC, então envia os dados comparados para o display LCD em forma de resposta ao código de barras lido. Se os dados da leitura coincidirem, em resposta, o PIC envia ao display o produto lido e seu valor. Caso o produto lido

¹² Armazenador de dados.

em forma de código de barras não conste na lista de produtos bufferizados, a resposta será “produto inválido”. Essa lista de produtos comentada é parte do programa gravado no microcontrolador.

Quando o programa é gravado no PIC, é gravada também a lista de todos os produtos do supermercado, ou seja, toda a lista de produtos é bufferizada previamente no microcontrolador para que o programa efetue as comparações necessárias.



Figura 4.4 – Caneta Ótica conectada na porta serial da placa de testes Acepic. Fonte: Do autor.

Para confecção de uma placa definitiva a ser acoplada na caneta ótica, o processo utilizado é o mesmo, mas desta vez utilizando apenas os dispositivos necessários para as funções da caneta ótica automática. Sendo assim, na confecção da placa definitiva a ser acoplada na caneta ótica, os dispositivos necessários são: porta serial, controladora da porta serial, socket do microcontrolador, microcontrolador PIC 16F648A, dois botões de comando, um LED para informar que a caneta ótica está ligada (LED aceso para ligada e apagado para desligada), porta controladora do display, display LCD de dezesseis pinos e quatro

linhas, entrada para fonte elétrica (ou bateria), conector para driver de pen drive (PenBS).

Para cada leitura de código de barras, o programa faz uma comparação com o dado lido decodificado pela caneta no seu banco de dados. Então busca a informação se o produto lido existe em seu cadastro ou não. Se existe, mostra na tela (LCD Out) o resultado da leitura (nome e valor do produto), caso não exista, mostra na tela a informação da inexistência do produto em seu cadastro, traduzido pelos caracteres “Produto Inválido”.

O compilador MikroC fornece a funcionalidade de transformar o código em linguagem C para linguagem hexadecimal. Dessa forma, o próprio compilador calcula o tamanho do programa e se o mesmo caberá ou não na memória do microcontrolador. Ao ser escolhido tal microcontrolador, por definição o programa não poderá ser maior do que a capacidade de memória deste PIC. Seguindo este procedimento, o programa foi desenvolvido obedecendo esta definição. Foi então escolhido para o protótipo o PIC 16F648A, que tem o dobro de memória do PIC utilizado anteriormente, conforme citado no Capítulo 3 deste trabalho.

Na figura 4.5 está ilustrada a compilação do programa em sua versão 2.8 para gravação no PIC 16F648A.

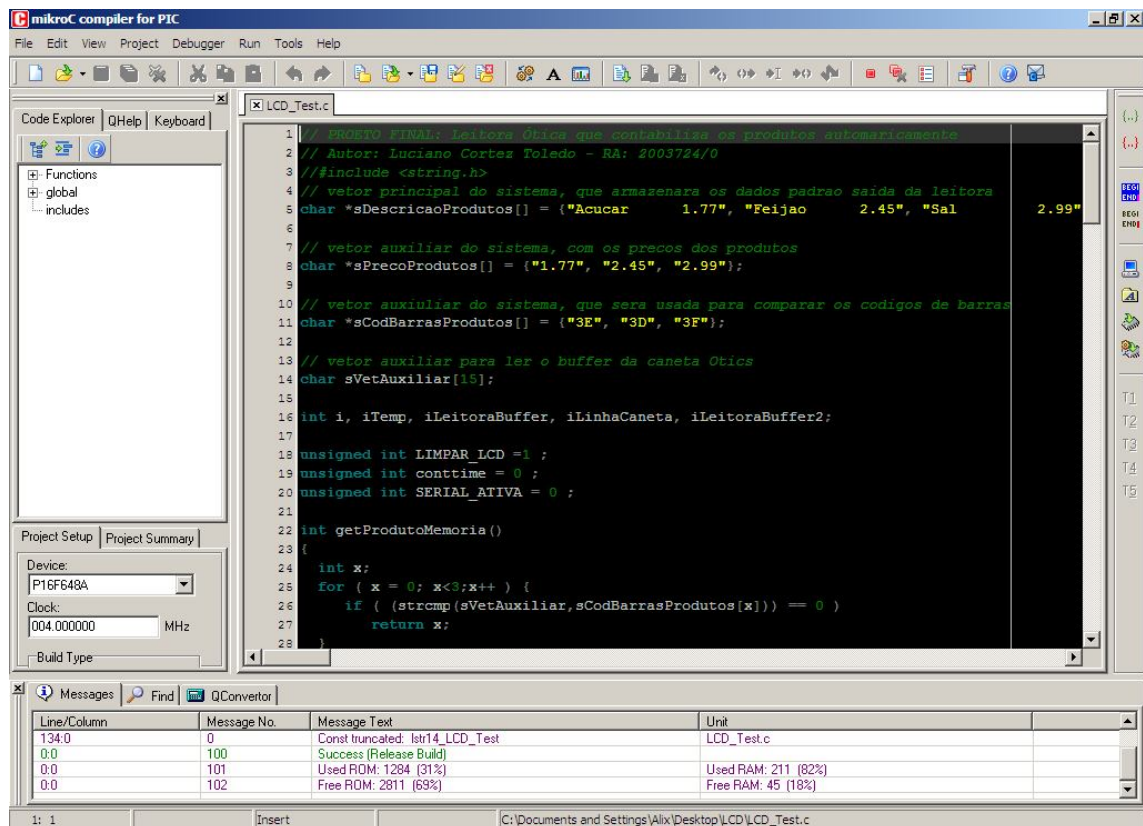


Figura 4.5 – Programa versão 2.8 compilado no MikroC. Fonte: Do autor.

Na figura 4.6, está ilustrada o programa na versão 2.8 transformado para linguagem hexadecimal, pronto para ser gravado no microcontrolador pelo aplicativo IC-Prog. Nota-se na imagem a escolha do PIC 16F648A para a gravação do programa no mesmo.

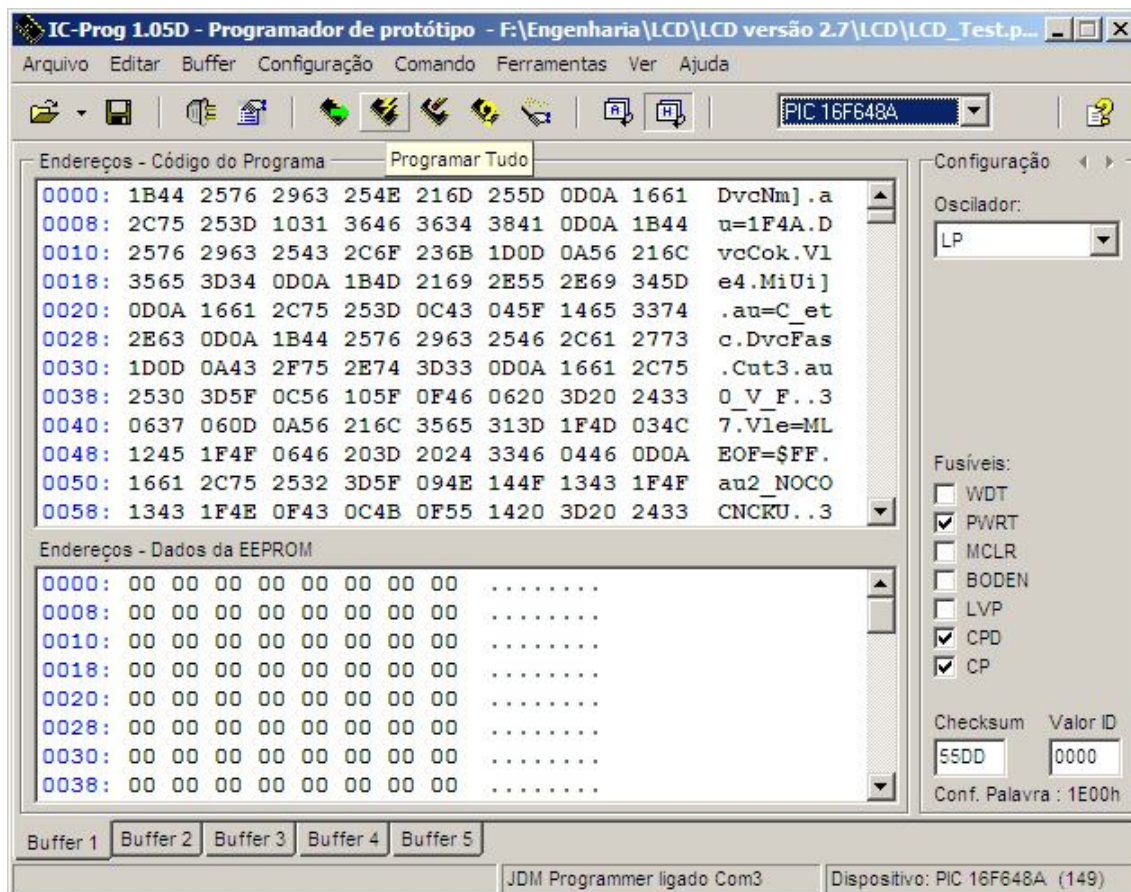


Figura 4.6 – Programa na versão 2.8 aberto no aplicativo IC-Prog a ser gravado no PIC 16F648A. Fonte: Do autor.

Com o programa gravado no PIC, efetua-se a execução do programa através do botão de reset da placa de desenvolvimento, conforme está ilustrada na figura 4.7.

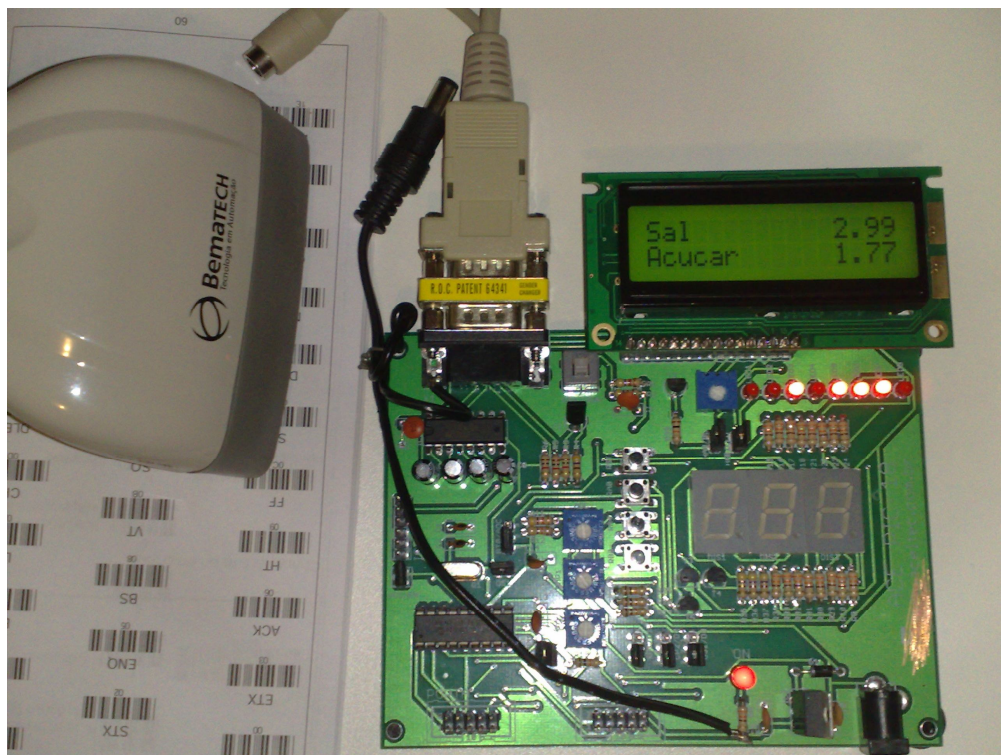


Figura 4.7 – Placa de desenvolvimento com o programa na versão 2.8 gravado e executado. Fonte: Do autor.

O kit está pronto para receber os dados lidos pela leitora de códigos de barras. Com a caneta ótica conectada na porta serial da placa de desenvolvimento, os códigos de barras lidos pela caneta são recebidos pelo microcontrolador. O programa faz a leitura dos dados, executa as funções comparativas e imprime na tela (display LCD) as devidas respostas. Um ponto importante a ser comentado é que a própria caneta faz a decodificação dos códigos de barras e envia para o programa os dados já convertidos em caracteres.

Para efetuar os testes de leitura de códigos de barras de produtos “reais”, faz-se necessária a explicação de pequena parte do código do sistema para o entendimento da decodificação dos códigos de barras em valores dos produtos. A seguir, temos um trecho do programa responsável por atribuir nomes e valores aos produtos, e as devidas explicações.

Código 1:

```
// vetor principal do sistema, que armazenara os dados padrao saida da leitora
char *sDescricaoProdutos[] = {"Acucar    1.77", "Feijao    2.45", "Sal      2.99",
                                "Produto invalido"};
```

```
//char *sDescricaoFirmawe[] = {"Caneta Otica" , "Versão : 2.8" };
// vetor auxiliar do sistema, com os precos dos produtos
//char *sPrecoProdutos[] = {"1.77", "2.45", "2.99"};

// vetor auxiliar do sistema, que sera usada para comparar os codigos de barras
//char *sCodBarrasProdutos[] = {"3E", "3D", "3F"};
```

Na última linha do trecho do programa, a representação de “3E”, “3D” e “3F” são os caracteres do código de barras do exemplo, ou seja, se a caneta ótica fizer a leitura do código de barras representada pela figura 4.8A, o resultado da leitura será o valor “3D”. Sendo assim, o código de barras exemplificado é representado pela figura 4.8B e seu valor no programa representado por “2.45”. Dessa forma, o produto de código de barras representado pela figura 4.8B tem seu valor igual a R\$ 2,45.



Figura 4.8A – Representação do código de barras dos caracteres 3D

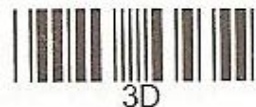


Figura 4.8B – Representação do código de barras dos caracteres 3D

Para os testes de leitura de códigos de barras de produtos reais, foram alterados no sistema os valores (caracteres) de exemplos do programa. Então, onde lia-se “3D” como valor do produto “feijão”, este valor foi alterado para o caracter real do produto, “7896101062016”. Efetuando o mesmo procedimento para outros produtos, o programa é compilado e gravado no PIC da placa de testes. Tem-se a leitura de teste “real” dos produtos ilustrada pela figura 4.9.



Figura 4.9 – Leitura do código de barras do produto “feijão”. Fonte: Do autor.

Tendo como base que os produtos cadastrados no sistema e que são lidos pelo programa são: “açúcar”, de valor igual a R\$ 1,77, “feijão”, de valor igual a R\$ 2,45 e “sal” de valor igual a R\$ 2,99, e que qualquer leitura de outros produtos não cadastrados no sistema, deverá se obter “Produto Inválido”, tem-se na figura 4.10 ilustração de teste de leitura de código de barras efetuado com um produto não cadastrado no sistema.

Estas ações são gerenciadas pela seguinte linha de comando do sistema:

Código 2:

```
char *sDescricaoProdutos[] = {"Acucar    1.77", "Feijao    2.45", "Sal        2.99",
                                "Produto invalido"};
```




Figura 4.10 – Leitura do código de barras do produto “amaciante”. Fonte: Do autor.

Para um melhor entendimento do programa desenvolvido para a solução, o subtópico 4.1 deste capítulo faz um detalhamento do sistema e de seu código fonte.

4.1 Explicação Detalhada do Código Fonte

4.1.1 Diagramação Gráfica

Apresenta-se a seguir uma representação gráfica do código-fonte detalhando o funcionamento do processo de captura e registro de produtos lidos pela caneta ótica. O diagrama a seguir é uma representação da máquina de estados do programa.

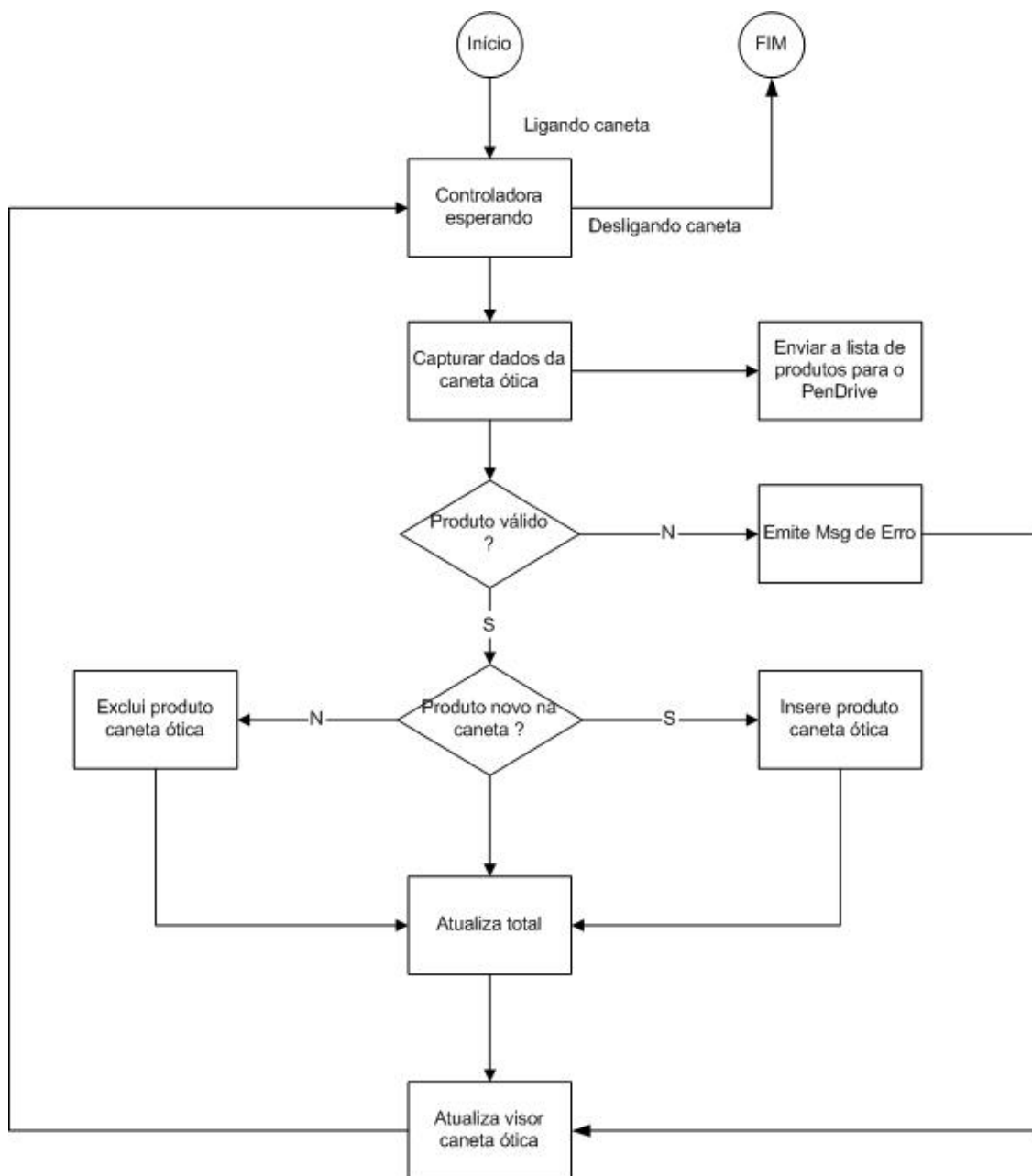


Figura 4.11 – Máquina de estados do programa da caneta ótica. Fonte: Do autor.

4.1.2 Principais Funções

Abaixo as principais funções do programa responsáveis por: inicializar a caneta ótica, capturar os dados da caneta ótica, analisar se o produto consta na

lista, inserir e retirar o produto da lista, exibir o produto no display LCD e enviar os dados formatados para o pen drive.

Função 1: inicializar a caneta ótica

```
Usart_Init(9600);  
void Tela_inicio()  
{  
    LCD_Write(2,3,"Caneta Otica");  
    LCD_Write(3,4,"Versao:2.8.2");  
    visor_inicio = 1;  
}
```

Esta linha de código inicializa a porta serial da caneta ótica. O número 9600 representa o bit interno requerido pela placa para o correto funcionamento. A figura 4.12 ilustra o hardware interagindo com esta parte do código.

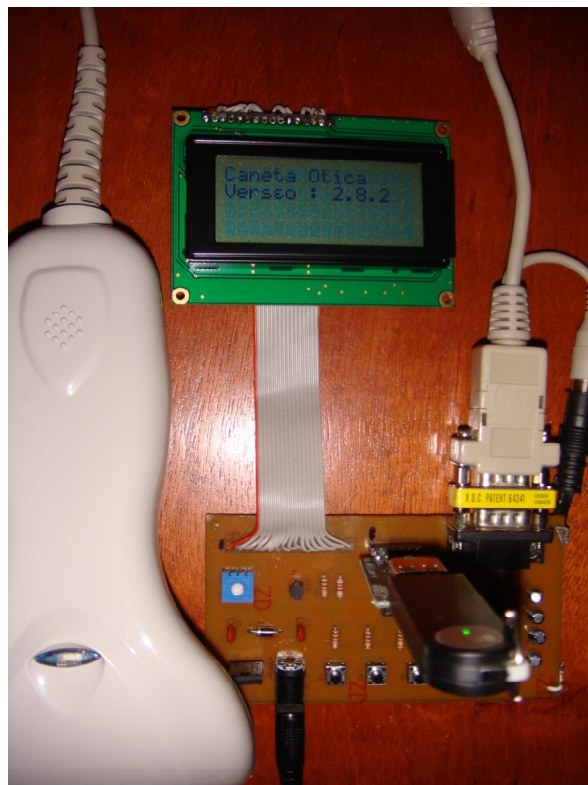


Figura 4.12 – Tela de inicialização da Caneta ótica. Fonte: Do autor.

Função 2: capturar os dados da caneta ótica.

```
void Nome_Arquivo_Serial() //LISTA.TXT
{
    Usart_Write(0x4C);    //L

    Usart_Write(0x69);    //I
    Usart_Write(0x73);    //S
    Usart_Write(0x74);    //T
    Usart_Write(0x41);    //A

    Usart_Write(0x2E);    //.
    Usart_Write(0x74);    //T
    Usart_Write(0x78);    //X
    Usart_Write(0x74);    //T
void Escrever_Produto_PenBS(unsigned short IdxProd,
                             unsigned short IdxPosicaoInicial,
                             unsigned short IdxPosicaoFinal)
{
    unsigned short x;
    //
    for (x=IdxPosicaoInicial; x < IdxPosicaoFinal; x++)
    {
        Usart_Write( asciiprodutos[x] );
    }

    //
    x = EEprom_Read(IdxProd);
    //
    x = ( x == 0xFF ) ? 0 : x;
    x = x + 0x30; // converto para ascii
    Usart_Write( x );
    Usart_Write( 0x0D );
void Esc_Caracter_Serial()
{
    Escrever_Produto_PenBS(0,0,9); //envia acucar definitivamente a PenDrive
    //
    Escrever_Produto_PenBS(1,9,18); //feijao definitivamente a PenDrive
    //
    Escrever_Produto_PenBS(2,18,27); //sal definitivamente a PenDrive
```

```

void Transmitir_Dados_Serial() //LISTA.TXT
{
    //unsigned short vlr;

    // COMANDO PARA ENVIAR DADOS
    Usart_Write(0x08);
    Usart_Write(0x20);
    // COMANDO PARA DIZER O TAMANHO DOS DADOS A SEREM ENVIADOS
    Usart_Write(0x00);
    Usart_Write(0x00);
    Usart_Write(0x00);
    Usart_Write(0x21);
    // COMANDO PARA FECHAR O COMANDO ANTERIOR
    Usart_Write(0x0D); //T
    // ESCRIBE DEFINITIVAMENTE NA PEN-DRIVE
    Esc_Caracter_Serial();
    //
    Usart_Write(0x0D); //T

```

Este trecho de código verifica, por meio da função `Usart_Data_Ready()`, se há qualquer buffer armazenado na caneta. Caso exista, a função retornará `TRUE`, a condicional será satisfeita e o programa armazenará o buffer na variável interna. Caso contrário retornará `FALSE` e o programa continuará esperando dados. A figura 4.13 ilustra a interação desta parte do código com o hardware.



Figura 4.13 – Tela de extração dos dados para o pen drive. Fonte: Do autor.

Função 3: analisar se o produto consta na lista.

```

if (visor_inicio==1 )
{
    LCD_Clr();
    Delay_ms(10);
    visor_inicio = 0;
}

if ((rec[0]=='3') && (rec[1]=='E'))
{
    msg1 = produto[0];/"Acucar";
    msg2 = preco [0];
    linha = 1;
    Trata_Produto(0, 177);
}
else
    if ((rec[0]=='3') && (rec[1]=='D'))
    {
        msg1 = produto[1];/"feijao";
        msg2 = preco [1];
        linha = 2;
        Trata_Produto(1, 245);
    }

```

```

    }
    else
    if ((rec[0]=='3') && (rec[1]=='F'))
    {
        msg1 = produto[2]; //"Sal";
        msg2 = preco [2];
        linha = 3;
        Trata_Produto(2, 299);
    }
    else
    {
        linha = 0;
    }
}

```

Este trecho de código verifica se o produto consta na matriz de produtos do sistema (representado pela variável *retira*). Caso o produto conste na lista, a função retornará TRUE e o produto deverá ser somado no total de produtos lidos e adicionado na matriz EEPROM (memória interna do microcontrolador). A figura 4.14 compara a execução do código (software) com a execução do hardware (display LCD).

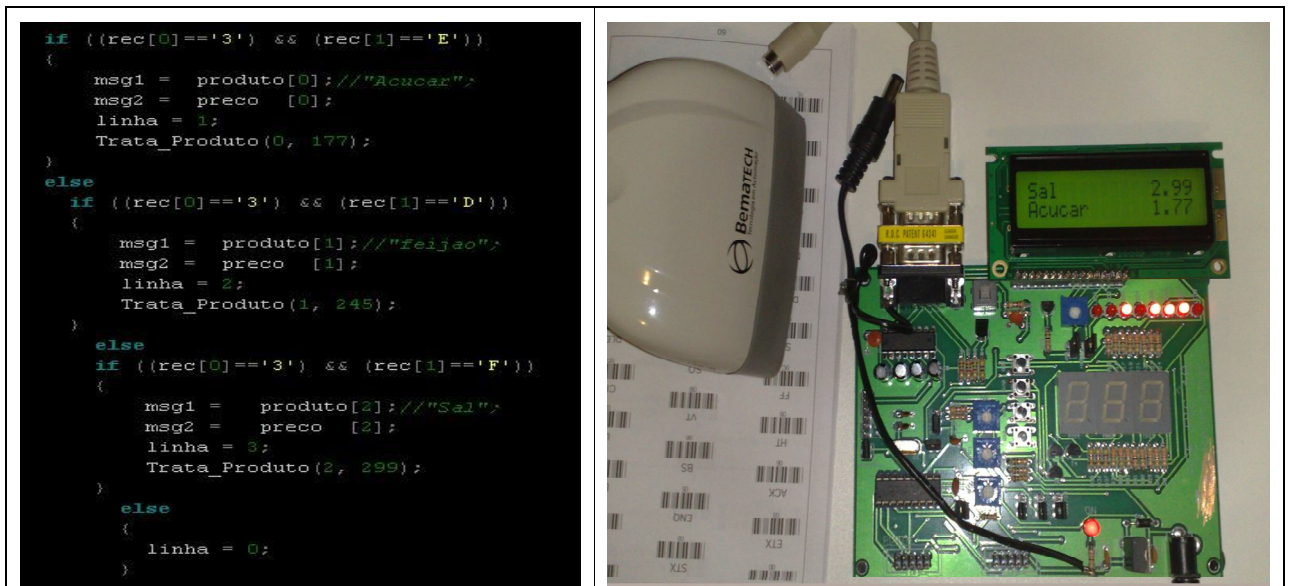


Figura 4.14 – Comparação entre execução do software no hardware. Fonte: Do Autor.

Função 4: inserir o produto na lista.

```
// Código para escrever na memória ROM
// Usado para banco de dados de produtos lidos na caneta
{
    qtde = ( qtde == 0xFF ) ? 1 : qtde + 1;
    vlr = vlr + preco;
    EEProm_Write(posicaoMemoria,qtde);
}
```

Este trecho de código registra o produto lido na matriz de produtos da caneta (memória EEPROM).

Função 5: retirar o produto da lista.

```
if ( retira == 1 )
{
    if ( qtde == 1 || qtde == 0xFF ) {
        msg1 = produto[4];
        msg2 = produto[5];
    }
    if ( qtde != 0xFF )
    {
        if ( qtde == 1 )
            qtde = 0xFF ;
        else
            qtde = qtde - 1;
        vlr = vlr - preco;
        EEProm_Write(posicaoMemoria,qtde);
    }
    retira = 0;
}
```

Este trecho de código retira o produto lido da matriz de produtos da caneta (memória EEPROM). A figura 4.15 ilustra o efeito do software no hardware, que no caso, aciona a ação “APAGAR” da lista de compras na próxima leitura de qualquer código de barras.



Figura 4.15 – Função “Apagar da lista compras” acionada. Fonte: Do autor.

Função 6: exibir o produto no display LCD.

```
// se leu um dos produtos registrados
if (linha > 0)
{
    // se retira o produto, limpa a linha do produto
    LCD_Write(linha,1,msg1);
    LCD_Write(linha,13,msg2);

    LCD_Write(4,1,"Valor:  ");
    vlrp = vlr / 1000;
    LCD_Ch(4,12,vlrp+0x30);
    vlrp = (vlr / 100) % 10;
    LCD_Ch(4,13,vlrp+0x30);
    LCD_Ch(4,14,',');
}
```

```

        vlrp = (vlr / 10) % 10;
        LCD_Ch(4,15,vlrp+0x30);
        vlrp = vlr % 10;
        LCD_Ch(4,16,vlrp+0x30);
    }
else
    LCD_Write(4,1,"Produto Invalido");

```

Este trecho de código exibe os produtos cadastrados na matriz de produtos da caneta. Ele mostra os produtos da linha 1 até a última linha 3 com os seus respectivos preços. O total será mostrado na linha 4. As funções responsáveis por mostrar o conteúdo no display LCD chamam-se *LCD_CH*(linha, coluna, caracter) e *LCD_Write*(linha, coluna, texto). A figura 4.16 ilustra as linhas de código citadas atuando no hardware (efetuado leitura de produtos).



Figura 4.16 – Código agindo no hardware (efetuando leituras dos códigos de barras dos produtos). Fonte: Do autor.

Função 7: enviar os dados formatados para o pen drive.

```
if(!PORTA.F5)
{
LCD_Clr();
Delay_ms(120);
LCD_Write(2,2,"EXTRAINDO DADOS");
LCD_Write(3,4,"AGUARDE...");
PenDrive();
}

// Funcao Para Extrair os dados para a PenDrive
void PenDrive()
{
FILE *arq=fopen("Lista.txt", "w");
//Escreve a lista no Pen Drive
for (prod=0;prod<3;prod++)
{
Usart_Write(0x0D); //CR
Usart_Write(0x08); //Habilita escrita
Usart_Write(0x20); //Espaço
Usart_Write(0x06); //Tamanho dos dados
Usart_Write(0x0D); //CR
Usart_Write(EEPROM_Read(prod)); //Dado a ser enviado
Usart_Write(0x0D); //CR
}
}
```

Este trecho de código é responsável por enviar os dados para o buffer do pen drive. A função `Usart_Write()` envia caracter a caracter para o buffer da caneta.

Com o programa recompilado já com a função acima descrita inserida no mesmo, o programa foi regravado no microcontrolador e executado, para o teste de extração de dados da caneta ótica para o pen drive (procedimento de recompilação do programa e gravação no PIC ilustrado nas figuras 4.5 e 4.6).

Na figura 4.17 está ilustrada a tela do programa recompilado, na versão 2.8, transformado em hexadecimal pelo aplicativo IC-Prog. Nesta versão, já está inclusa a no código função de extração de dados da caneta ótica (por meio da placa PenBS - porta USB).

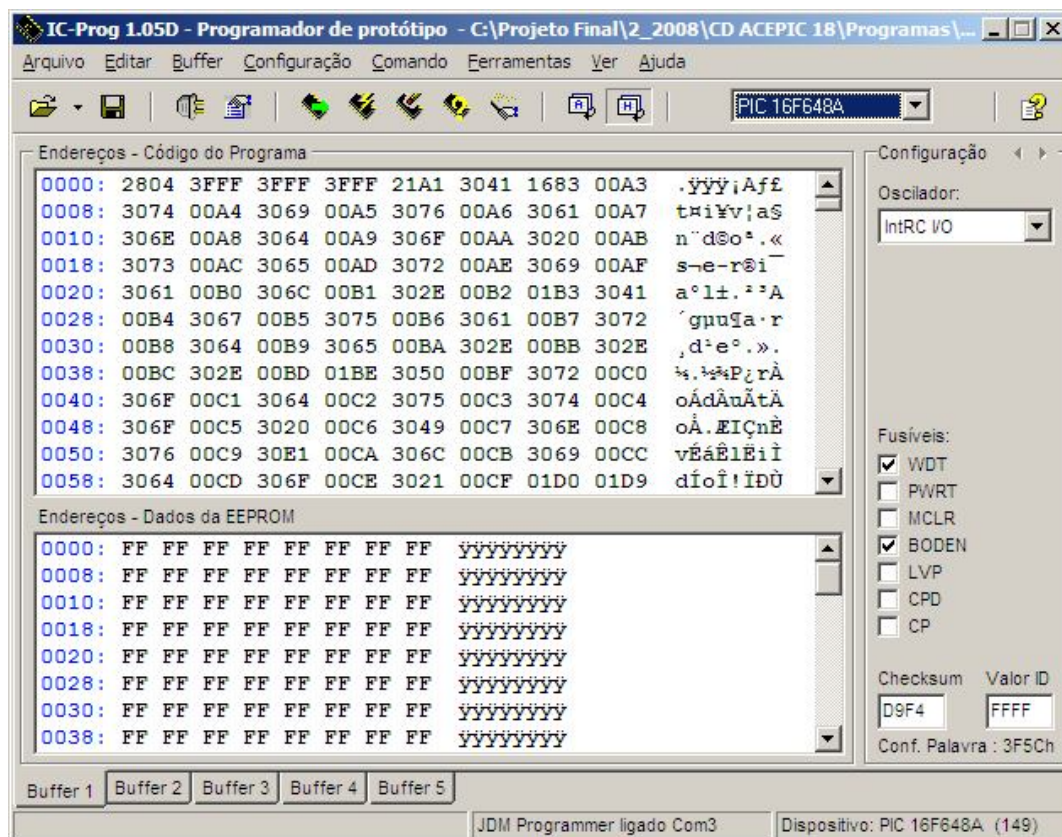


Figura 4.17 – Código do programa na versão 2.8 em transformado em hexadecimal pelo aplicativo IC-Prog. Fonte: Do autor.

Após a nova versão do programa ser gravada no microcontrolador e executada pelo botão de execução da placa de desenvolvimento, executa-se a extração dos dados por meio do botão de comando da placa. Dessa forma, o sistema extrai os dados lidos da pen drive conectada na placa PenBS, enquanto mostra no display a informação de que os dados estão sendo extraídos. Uma ilustração deste procedimento está ilustrada na figura 4.18. Para a execução deste, a placa PenBS foi conectada à placa de desenvolvimento pela Porta B¹³ desta.

¹³ Porta para conexão de dispositivos na placa de desenvolvimento.



Figura 4.18 – Extração de dados da caneta ótica. Fonte: Do Autor.

Após este último teste do projeto, verificou-se o sucesso na execução do código da caneta ótica, onde o mesmo informou na tela (display) a execução de extração dos dados da caneta para o pen drive. Conforme explicado no trabalho, a os dados extraídos são a lista de compras e os valores dos produtos, assim como o total geral das compras. Esses dados são formatados pelo código, em um arquivo txt gravado no pen drive, que, de acordo com o projeto, é o objeto final do trabalho.

De posse destes resultados, foi então confeccionado o lay-out para a placa definitiva acoplada na caneta ótica. Foram enviadas as informações necessárias para a empresa Acepic confeccionar tal placa. Tais informações são os dispositivos necessários na placa eletrônica que fazem com que o projeto possa operar junto à

caneta ótica convencional. Tais dispositivos são: porta serial, controladora de porta serial, socket para microcontrolador, microcontrolador da família 16F648A, display LCD digital de quatro linhas, porta controladora do display, LED's, botões de comando, driver para pen drive (placa PenBS) e entrada de corrente elétrica. Na figura 4.19 é ilustrado o lay-out da placa definitiva, confeccionado pelo fabricante Acepic. O esquema elétrico desta placa está ilustrado no Anexo II deste trabalho. Por meio desta, verifica-se como foi projetado o esquema elétrico desta placa.

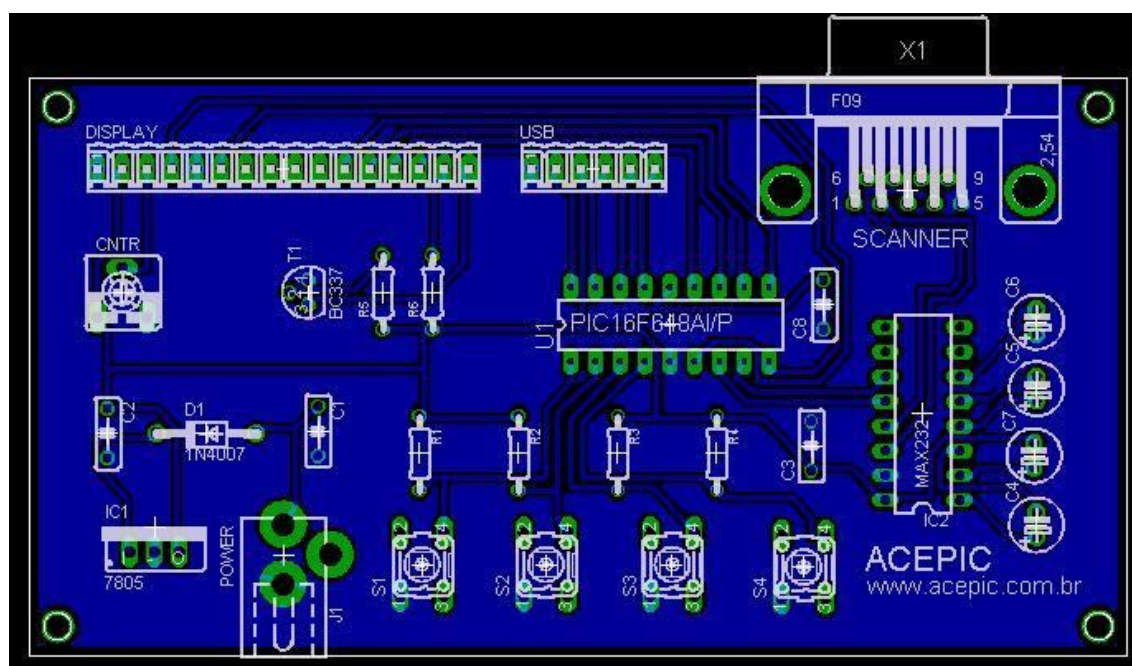


Figura 4.19 – Lay-Out da placa definitiva do projeto Caneta Ótica. Fonte: Do autor.

A pinagem da placa definitiva foi efetuada obedecendo ao esquema elétrico da placa ilustrada pela figura 4.20, tendo-se os pinos B7, B6, B5, B4, B3 e B0 do microcontrolador gerenciando a porta do display, os pinos A6, A7, A0 e A1 gerenciando a porta USB, os pinos B1 e B2 gerenciando a porta serial, os pinos A2, A3, A4 e A5 gerenciando os botões de comando. A placa possui uma alimentação de 12 V, sendo convertida para 5 V pelo conversor de tensão (situado próximo à entrada de alimentação) e quatro capacitores auxiliares. Possui também um dispositivo de resistência regulável para contraste da imagem do display LCD e resistores de auxílio. Tal dispositivo é chamado de potenciômetro. Além dos botões

de comando ligados aos pinos do microcontrolador, também com resistores auxiliares, a placa também possui um chip RS232 responsável pela troca de dados binários entre um DTE ¹⁴ e um DCE¹⁵. É comumente utilizado em portas seriais de PC's. Na placa definitiva, é responsável pela comunicação da porta serial com o microcontrolador.

As informações necessárias para a produção da placa definitiva foram: trilhas de comunicação e gerência do microcontrolador às portas seriais do scanner e da placa PenBS; produção de botões para execução de funções do código, com conexões simultâneas (trilhas elétricas) com os pinos do microcontrolador e pinos da porta paralela do display LCD; fonte de convergência de voltagem de 220V para 5V; porta paralela para conexão do display LCD e trilhas elétricas para gerência dos pinos do display LCD; trilhas elétricas para comunicação Tx/Rx simultâneas com as portas seriais (scanner e placa PenBS), porta paralela do display LCD e pinos do microcontrolador; potenciometro para regulagem do *break-light* ¹⁶ do display LCD.

Pelo circuito elétrico, o microcontrolador efetua a gerência das portas seriais do scanner e da placa PenBS por comunicação serial, efetuada pelas trilhas elétricas da placa. Os pinos B1 e B2 do microcontrolador são responsáveis pela comunicação serial tanto da porta serial do scanner quanto da porta serial da placa PenBS. Para tanto, foi efetuada uma comunicação serial do microcontrolador à porta da placa PenBS e descartando a comunicação via IC2P, raspando a trilha responsável por tal comunicação e conectando fios de cobre na comunicação TX do microcontrolador ao RX da porta da placa PenBS, e vice-versa. Desse modo, quando estabelece uma comunicação com uma das portas seriais, a comunicação da outra é interrompida.

O esquema elétrico da placa definitiva foi desenhado para que todos os pinos do microcontrolador se comuniquem simultaneamente com todos os outros pinos das portas seriais, assim como com os botões de execução de funções, uma vez as funções do código se estabelecerem ao mesmo tempo, ou seja, todas as funções do sistema do microcontrolador executam funções simultaneas, não havendo parada de uma função à espera de outra, fazendo que os eletrônicos não dependam dos outros para operarem.

¹⁴ Terminal de dados.

¹⁵ Comunicador de dados.

¹⁶ Luz interna do LCD.

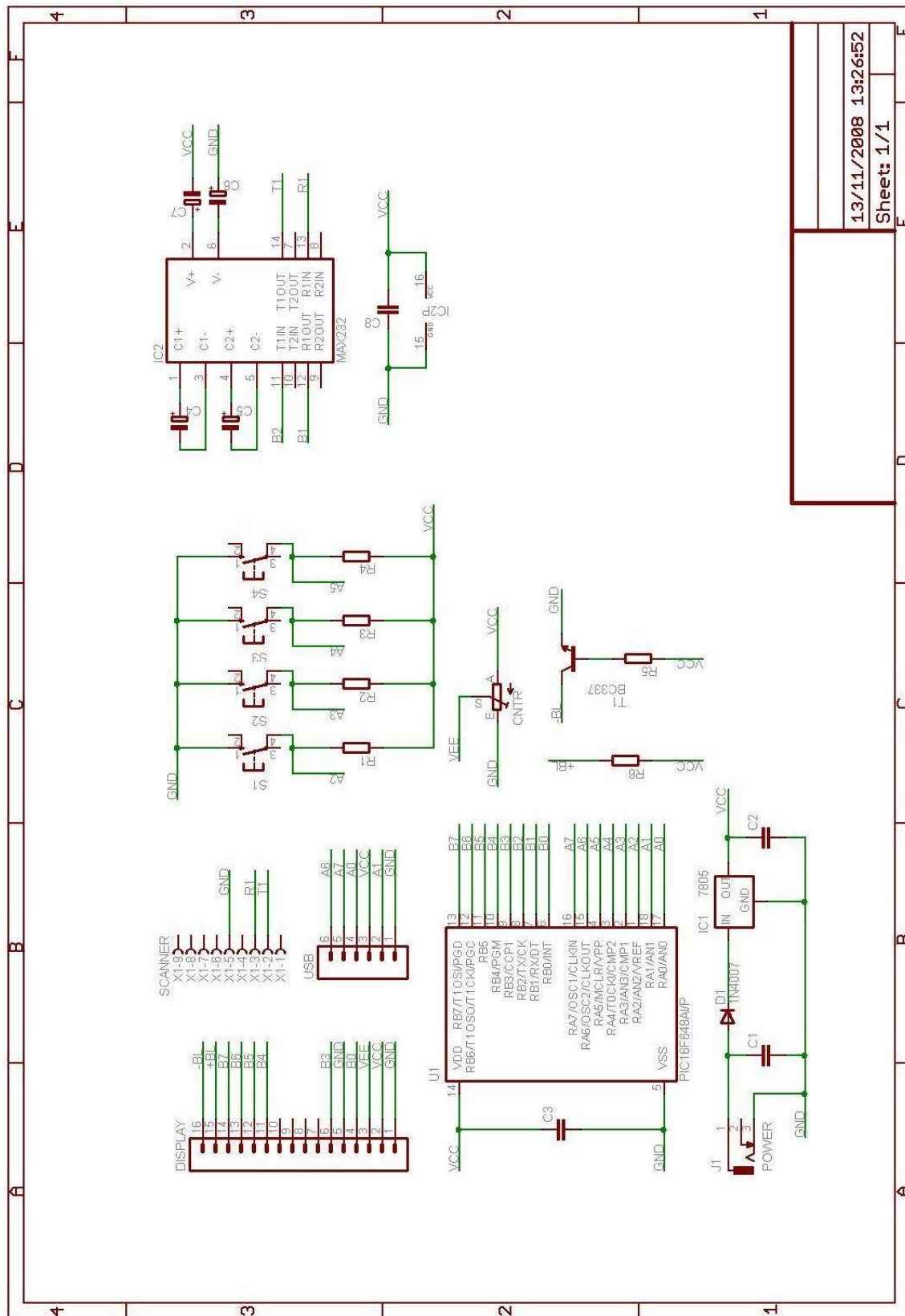


Figura 4.20 – Esquema elétrico da placa definitiva. Fonte: Acepic

A figura 4.21 ilustra a placa definitiva da caneta ótica executando o programa em sua versão 2.8.2 (programa recompilado para gravação no microcontrolador da placa definitiva).

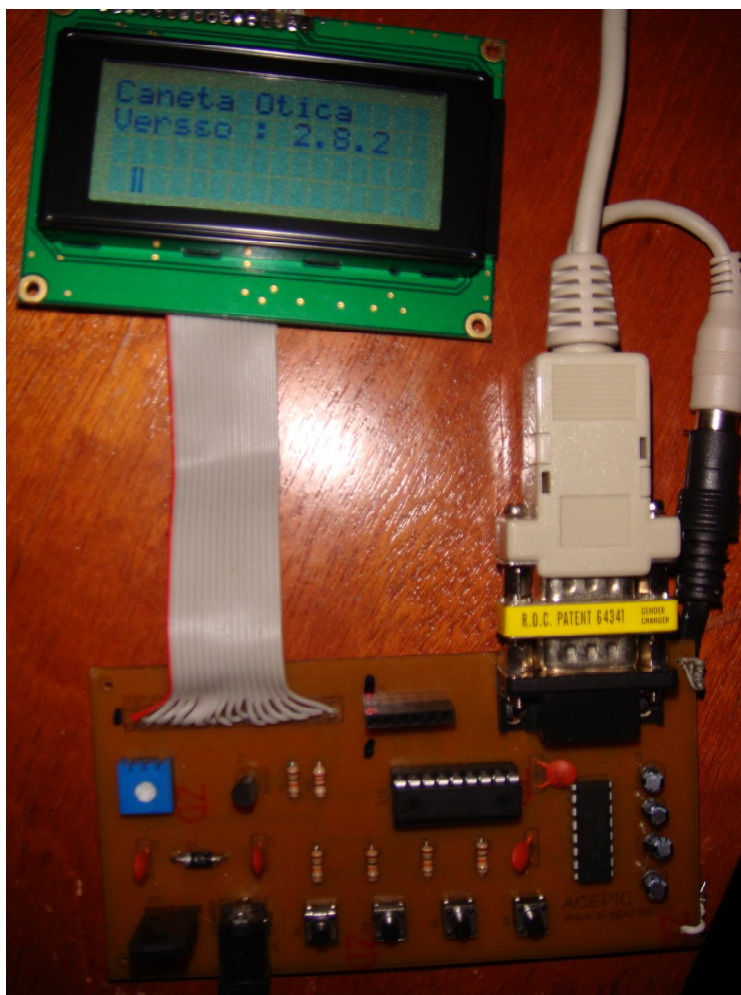


Figura 4.21 – Imagem da placa definitiva da solução Caneta Ótica.
Fonte: Do autor.

A figura 4.22 ilustra a placa definitiva da caneta ótica já com a placa PenBS e o pen drive conectados..

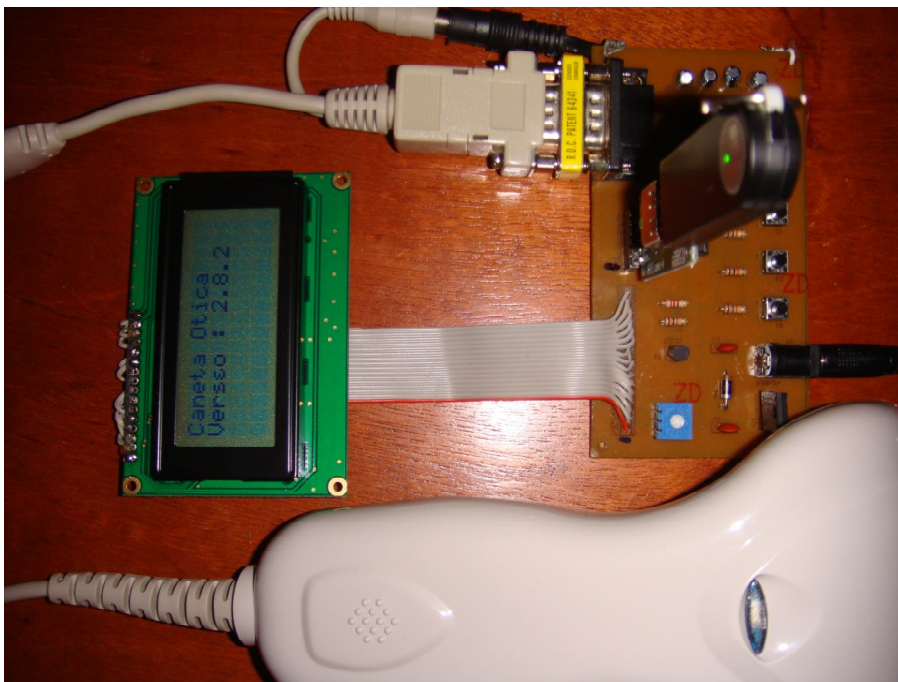


Figura 4.22 – Placa definitiva com placa PenBS e pen drive. Fonte: Do autor.

4.2 Testes Utilizando a Placa Definitiva da Caneta Ótica

Com o programa recompilado em sua versão 2.8.2, foram feitos os testes de leitura de códigos de barras em produtos “reais” utilizando os conceitos sobre os códigos de barras explicados anteriormente. Para isso, foi feito o processo inverso do anterior, ou seja, foram mantidos os códigos “3E”, “3D” e “3F” no programa, referentes aos nomes e valores dos produtos, ao invés das numerações reais, pois geraram estouro de memória do microcontrolador ao recompilar o programa. Para a leitura pela caneta ótica, os códigos de barras dos produtos foram substituídos pelos códigos comentados, assim, o programa foi re-compilado e gravado no microcontrolador. Com isso, deu-se início aos testes de leituras de códigos de barras utilizando a placa definitiva da caneta ótica.

Na figura 4.23 é ilustrado o teste de leitura para o produto “açúcar”.



Figura 4.23 – Teste de leitura de código de barras do produto “açúcar”.

Fonte: Do autor

Na figura 4.23 é ilustrado o teste de leitura para o produto “macarrão”. Este é um teste para um produto não cadastrado, portanto, o sistema mostra no display a informação de produto inválido.



Figura 4.24 – Teste de leitura de código de barras do produto “macarrão”. Fonte: Do autor.

Após os produtos serem lidos, tem-se a informação do valor do total da contabilidade dos produtos no display LCD. Esta informação também está contida no programa, portanto, pode-se extrair os dados dos produtos lidos, é a lista das leituras, no caso da solução, a lista de compras do cliente. O próximo teste é a extração da lista de compras, ou seja, extração da lista dos produtos lidos pela caneta ótica e armazenado no programa. O teste é feito por meio do botão de comando para extração dos dados, presente na placa definitiva da caneta ótica. A qualquer momento o cliente do supermercado poderá apertar este botão que os dados serão extraídos para o pen drive. Na figura 4.25 é ilustrado o teste de extração de dados do sistema para o pen drive.



Figura 4.25 – Extração de dados do sistema para o pen drive. Fonte: Do autor

Após a extração dos dados, pode-se ter acesso à lista de compras conectando o pen drive no PC. Abrindo o arquivo, tem-se toda a lista dos produtos lidos, seus valores e o total da contabilização dos produtos. Na figura 4.26 está ilustrado a lista dos produtos lidos pela caneta ótica, exemplificando o acesso à lista de compras do cliente.

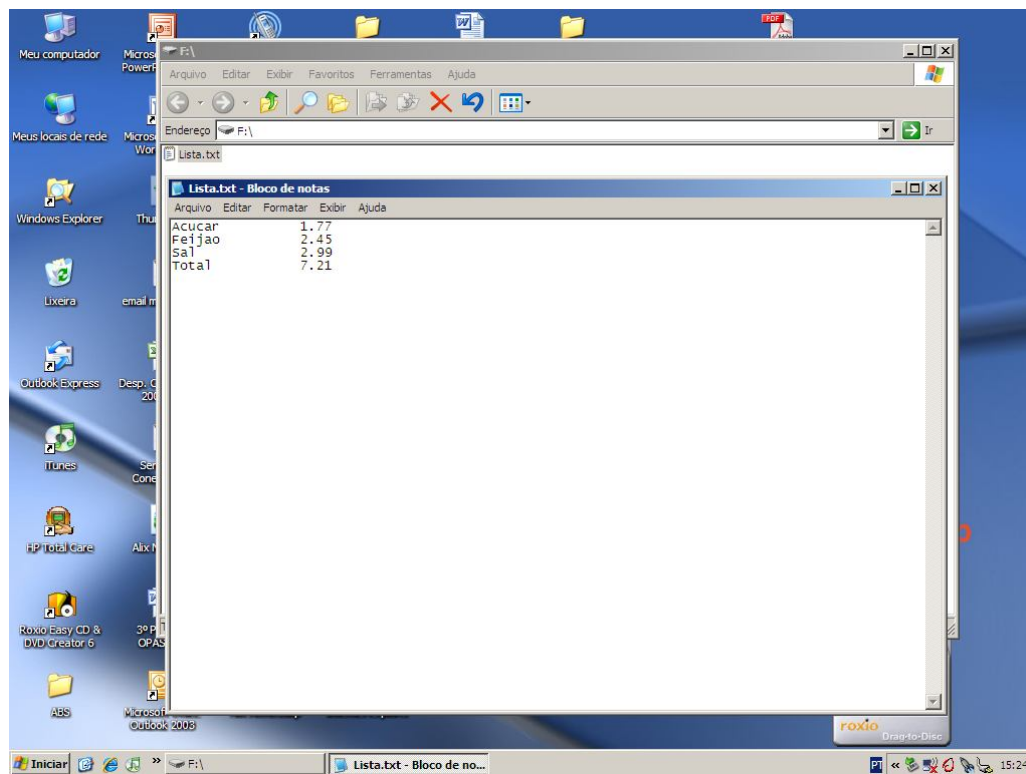


Figura 4.26 – Lista dos produtos lidos pela caneta ótica. Fonte: Do autor.

Após os testes válidos da solução e a certeza do produto funcionando, têm-se no Capítulo 5 a seguir as descrições da aplicabilidade da solução em modo acadêmico e algumas explanações da aplicabilidade no modo real. A figuras do Capítulo 5 comprovam a operabilidade da solução, ilustrando os testes de leitura dos códigos de barras dos produtos e a execução simultânea da contabilidade dos produtos lidos, ou seja, em tempo real tem-se sempre o valor parcial de todas as leituras. Conforme explicado anteriormente, o valor parcial transforma-se em valor total quando o botão de comando de extração dos dados é pressionado e os dados transmitidos ao pen drive. Consta-se então no arquivo extraído o valor total das leituras efetuadas.

Capítulo 5. APLICAÇÃO DA SOLUÇÃO

Sabe-se que toda solução encontrada e aplicada implica em tomada de ações e estas requerem recursos que devem ser previstos: material, tempo e dinheiro. O objetivo é a aplicação da solução e a manutenção do processo sob controle nos novos níveis projetados, com estabelecimento das informações necessárias às pessoas que vão aferir e controlar o processo nessas novas condições.

5.1 Laboratório de Experimentação

Foram coletados 20 produtos para a leitura do código de barras com a análise do tempo médio gasto durante o processo. Têm-se abaixo duas imagens coletadas durante o laboratório de experimentação. Na figura 5.1, é ilustrado a leitura do produto “feijão”, que consta na lista de produtos da caneta.



Figura 5.1 – Teste de leitura de código de barras do produto “feijão”.

Fonte: Do autor

Na figura 5.2 é ilustrado o teste de leitura para o produto “sal”.

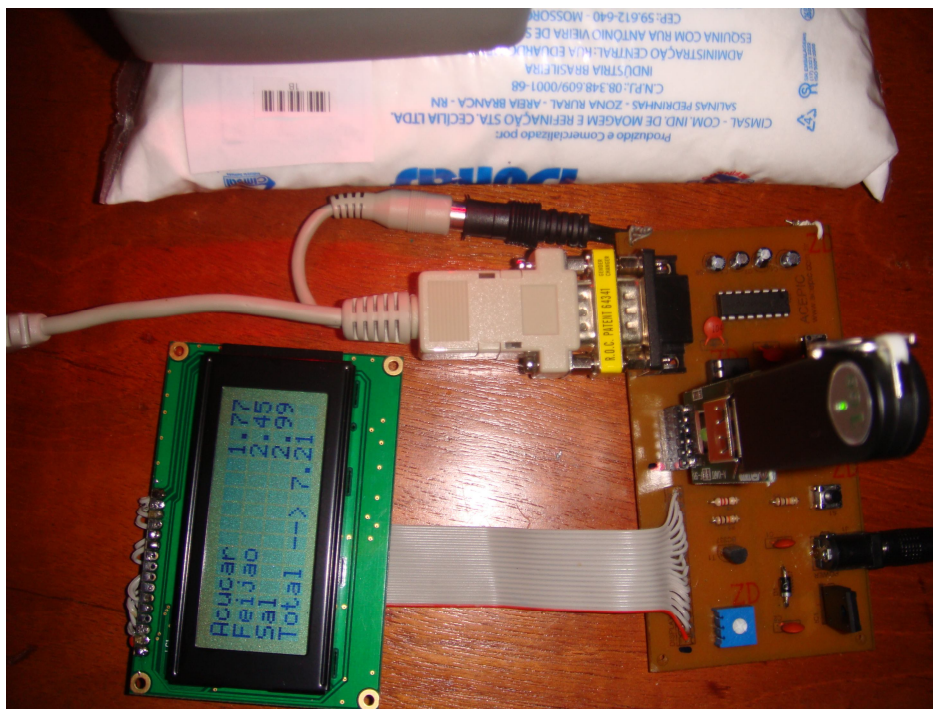


Figura 5.2 – Teste de leitura de código de barras do produto “sal”.
Fonte: Do autor.

Nota-se neste teste a leitura do terceiro produto da lista, os valores dos produtos lidos anteriormente e o valor total das leituras dos códigos de barras dos produtos. A partir disso, os dados podem ser extraídos conforme explanado no Capítulo anterior e ilustrado nas figuras 4.18 e 4.19.

5.2 Resultados Obtidos

Observou-se que o uso da caneta ótica atende efetivamente a solução para reduzir o tempo de permanência dos clientes nas filas de caixas em estabelecimentos comerciais. A análise custo-benefício comprova a eficiência, eficácia e efetividade da solução desenvolvida, aplicando exemplos reais dos

tempos gastos em filas, comparando o método tradicional com a nova solução apresentada. Os resultados asseguram o sucesso da sua aplicabilidade.

5.3 Variáveis de Impacto Indireto no Projeto

Academicamente o projeto é válido. Porém, para a utilização prática da caneta ótica automatizada é preciso olhar para as variáveis de dispersão existentes no conjunto de medidas que se pode implementar, verificando a correlação entre elas: tempo, custo, melhoria da qualidade do processo de vendas, etc.

Todos os problemas de um negócio giram em torno, ou é uma consequência direta, de três fatores: aprimorar os níveis de qualidade, reduzir o ciclo do tempo e otimizar os custos de produção.

5.4 Resultados Obtidos das Comparações Efetuadas

Durante a fase de validação do projeto foram testados 20 produtos, tendo-se os códigos de barras nesses produtos, a caneta ótica. O tempo gasto para a leitura dos 20 produtos foi de 1 minuto.

No supermercado da forma tradicional, o tempo gasto para as compras, sem considerar o tempo de 12 minutos para escolha dos 20 produtos, foi de 33 minutos (22 minutos na fila, mais 3 minutos para retirada das compras do carrinho, 4 minutos para a leitura no caixa, 2 minutos para o pagamento e 2 minutos para empacotamento)*¹⁷.

Considerando que o tempo de 22 minutos na fila corresponde a 3 pessoas antes da entrega efetiva do pen drive ao Caixa, e que, o processo de compras usando o pen drive é de 1 minuto, tem-se que o mesmo tempo de espera é de 3 minutos até a entrega efetiva do pen drive. Resultado: 33 minutos até a efetiva

¹⁷ *. Tempo real gasto no supermercado Carrefour, às 17h30min do dia 05/11/2008.

saída do supermercado no modo convencional e 4 minutos até a efetiva saída do laboratório de experimentação.

Aplicando-se a regra matemática onde 33 representa 100% do tempo gasto com as compras na forma convencional e 4 representa x% do mesmo tempo que será levado na forma proposta, tem-se a seguinte fórmula: $x = 4 \cdot 100 / 33$ (regra de 3 convencional). Portanto, x é igual a 12,12% de tempo gasto com as compras utilizando a nova forma proposta. Isso representa um ganho de aproximadamente 88% do tempo utilizado para fazer compras.

Capítulo 6. CONCLUSÃO

Este projeto foi concebido visando minimizar o tempo de espera em filas, por parte do cliente, em estabelecimentos comerciais, dar pleno conhecimento ao cliente sobre o preço do produto no ato da compra, permitindo-lhe tomar decisões também em tempo real, agilizando o processo de pagamento por parte do estabelecimento comercial, gerando conforto maior ao cliente final e dando maior visibilidade na projeção do faturamento do dia.

Para isto, neste trabalho foi apresentada uma alternativa para minimizar os problemas acima mencionados: uma caneta ótica, capaz de registrar e contabilizar automaticamente o nome do produto e seu preço via código de barras, enquanto simultaneamente exibe na tela da caneta o nome e valor do produto, bem como o somatório de todos os valores lidos. Ao término das compras, o instrumento é capaz de transferir os dados registrados para o computador central num ato contínuo o cliente recebe a fatura.

Esse projeto apresentou a tecnologia então existente para desenvolvimento de um hardware mais avançado, e o desenvolvimento de um software na linguagem de programação, oferecendo mecanismos para se planejar e gerenciar o processo de desenvolvimento de um sistema de informação, sob os pontos técnicos e econômicos.

Feita a simulação do protótipo, fica indicado que, tecnicamente, o produto encontra-se passível de desempenhar as funções da forma como planejada, identificando a adequação dos equipamentos, dos materiais e do ambiente de trabalho. O problema dos erros de configuração dos dispositivos pode ser sanado no decorrer do processo de melhoria e aperfeiçoamento da caneta ótica.

Para a implementação completa do produto de forma comercial, faz-se necessário um maior tempo de estudo para que se resolvam problemas técnicos encontrados durante o desenvolvimento deste projeto. A primeira parte do projeto, apresentada academicamente, não visa resolver fraudes que possam existir no uso da solução, como por exemplo: 1- o cliente do supermercado tentar abrir o arquivo final em um notebook para, supostamente, apagar itens da lista de compras; 2- o

cliente desmagnetizar o produto e tentar levá-lo sem efetuar o pagamento; 3- o cliente furtar produtos já lidos e desmagnetizados de outros carrinhos de compras; 4- o cliente tentar sair do estabelecimento levando consigo a caneta ótica e o produto desmagnetizado; 5- o cliente retirar as etiquetas de código de barras dos produtos; entre outros, que possam ocorrer. Entende-se que fraudes e ações anti-furto dentro dos estabelecimentos são de inteira responsabilidade dos mesmos.

No próximo tópico deste trabalho, estão sugeridas, como projetos futuros para o mesmo, algumas soluções que podem resolver os problemas citados no parágrafo anterior (tentativas de fraudes da solução por parte de clientes e estabelecimentos comerciais). Algumas das sugestões já fazem parte dos processos seguintes deste projeto, e podem ser trabalhadas em parceria com os próprios estabelecimentos comerciais. Para tanto, faz-se necessário algum tempo de pesquisa das soluções que poderão ser utilizadas para a viabilização comercial do projeto, como por exemplo, utilização da tecnologia Bluetooth e de rede de dados do tipo Wi-Fi.

6.1 Sugestões para Trabalhos Futuros

Como sugestão para trabalhos futuros:

1. Implementar a solução utilizando rede Wi-Fi, de modo que a cada produto lido, os dados sejam enviados ao computador central. Isso impossibilita fraudes do tipo de acesso à lista final, ou seja, de maneira alguma clientes teriam qualquer tipo de acesso à lista de compras que não o mostrado no display LCD.

2. Implementar extração dos dados finais (lista das compras e valor total) por meio da tecnologia Bluetooth (da caneta ótica diretamente para o computador do caixa).

3. Implementação da solução com utilização de uma memória externa acoplada na placa. Dessa forma, haveria uma capacidade maior de *bufferização* de produtos no sistema.

4. Implementar o envio dos dados seriais da caneta ótica, por meio de SMS ou outro meio de comunicação, para um computador central, a cada leitura de códigos de barras pela caneta ótica utilizada. Esse procedimento enviabilizará qualquer troca de produtos já lidos dentro do supermercado, ou o abandono da caneta ótica por parte do cliente (para alguma possível tentativa de sair com algum produto desmagnetizado de dentro do supermercado).

5. Implementação de uma interface amigável por onde os produtos sejam inseridos no sistema. Por meio de um software o supermercado inserirá sua lista de produtos (caracteres dos códigos de barras, nomes dos produtos e valores do produto). Esta lista deverá ser armazenada em um banco de dados e não mais no próprio sistema.

REFERÊNCIAS

[IBM 1]

Site da IBM disponível em <http://www-03.ibm.com/e-business/br>, último acesso em 15/09/2008.

[ABE 2]

Site da ABEPRO (*Associação Brasileira de Engenharia de Produção*) disponível em <http://www.abepro.org.br>, último acesso em 25/09/2008.

[ABE 3]

Site da ABEPRO (*Associação Brasileira de Engenharia de Produção*) disponível em <http://www.abepro.org.br>, último acesso em 25/09/2008.

[ABE 4]

Site da ABEPRO (*Associação Brasileira de Engenharia de Produção*) disponível em <http://www.abepro.org.br>, último acesso em 25/09/2008.

[IBM 5]

Site da IBM disponível em <http://www-03.ibm.com/e-business/br>, último acesso em 26/09/2008.

[IBM 6]

Site da IBM disponível em <http://www-03.ibm.com/e-business/br>, último acesso em 26/09/2008.

[FOL 7]

Site Do jornal Folha do Estado, disponível em <http://www.folhadoestado.net>, último acesso em 27/09/2008.

[FOL 8]

Site Do jornal Folha do Estado, disponível em <http://www.folhadoestado.net>, último acesso em 27/09/2008.

[FOL 9]

Site Do jornal Folha do Estado, disponível em <http://www.folhadoestado.net>, último acesso em 27/09/2008.

[ABE 10]

Site da ABEPRO (*Associação Brasileira de Engenharia de Produção*) disponível em <http://www.abepro.org.br>, último acesso em 25/09/2008.

[ABE 11]

Site da ABEPRO (*Associação Brasileira de Engenharia de Produção, estudo feito por MORABITO e LIMA, 2000*) disponível em http://www.abepro.org.br/biblioteca/ENESEP2000_E0184.PDF, último acesso em 25/09/2008.

[JOR 12]

Site do Jornal Livre, disponível em <http://www.jornallivre.com.br>, último acesso em 28/09/2008.

[MAR 13]

Site sobre PIC`s, disponível em <http://www.marcelo.perotto.nom.br/>, último acesso em 08/10/2008.

[JOR 14]

Site do Jornal Livre, disponível em <http://www.jornallivre.com.br>, último acesso em 28/09/2008.

[MIC 15]

Pereira, Fábio. Microcontroladores PIC. Segunda Edição. Editora Érica, 2003

[MIC 16]

Pereira, Fábio. Microcontroladores PIC. Segunda Edição. Editora Érica, 2003

[PIC17]

Zanco, Wagner da Silva. Microcontroladores PIC. Primeira Edição. Editora Érica, 2005.

[PIC18]

Zanco, Wagner da Silva. Microcontroladores PIC. Primeira Edição. Editora Érica, 2005.

[GS1 19]

Site Bar Codes Symbology, disponível em <http://www.gs1tw.org/twct/g1w>, último acesso em 10/10/2008.

[PAI 20]

Paixão, Silvério. Código GS1-128 – Uma Ferramenta Facilitadora da Rastreabilidade. Segurança e Qualidade Alimentar. Lisboa, 2006, disponível em <http://www.infoqualidade.net/SEQUALI/PDF-SEQUALI-01/n01-pg31-32.pdf>, último acesso em 10/10/2008.

Apêndice I

Estrutura do Trabalho

Para criar a estrutura do trabalho, foram usados padrões de projeto definidos pelas práticas pregadas pelo PMI (*Project Management Institute*), onde teremos as seguintes fases:

- **Iniciação:** foi representada pela Proposta de Projeto Final, aprovada pela instituição;
- **Planejamento:** desenvolvimento do escopo do projeto, seqüenciamento de atividades e definição de cronograma de projeto. Abaixo, o planejamento representado nas figuras 1.3 e 1.4.
- **Execução:** desenvolvimento da monografia e da caneta ótica;
- **Controle:** verificação se a execução está dentro do escopo e prazo definidos. Correção de rotas para sincronismo com o cronograma planejado e/ou redefinição de prazos, se necessário;
- **Fechamento:** apresentação do projeto (monografia + produto-solução) em banca de Projeto Final.


















		Nome da tarefa	Duração	Início	Término
1		 Projeto Caneta Ótica	166 hrs	Qui 04/09/08 20:00	Dom 12/10/08 18:00
2		 1a Sequencia (Hardware)	56 hrs	Qui 04/09/08 20:00	Ter 16/09/08 23:00
3		Aquisicao Peças (pedido pelo Correio)	40 hrs	Qui 04/09/08 20:00	Sáb 13/09/08 18:30
4		Aquisição do KIT de desenvolvimento	40 hrs	Qui 04/09/08 20:00	Sáb 13/09/08 18:30
5		montagem do protótipo no ProtoBoard	8 hrs	Sáb 13/09/08 18:30	Dom 14/09/08 18:00
6		integração de todos os componentes no PIC	8 hrs	Dom 14/09/08 18:00	Ter 16/09/08 23:00
7		 2a Sequencia (Software)	56 hrs	Qua 17/09/08 20:00	Seg 29/09/08 23:00
8		 desenvolvimento lib. de LCD	16 hrs	Qua 17/09/08 20:00	Dom 21/09/08 10:30
9		baixar lib da internet	3 hrs	Qua 17/09/08 20:00	Qua 17/09/08 23:00
10		Adequar a lib ao projeto	16 hrs	Qua 17/09/08 20:00	Dom 21/09/08 10:30
11		 desenvolvimento caneta c/ PIC	12 hrs	Dom 21/09/08 10:30	Ter 23/09/08 21:30
12		Entender protocolo serial de comunicação	12 hrs	Dom 21/09/08 10:30	Ter 23/09/08 21:30
13		criar rotina de leitura e quebra dos dados	4 hrs	Dom 21/09/08 10:30	Dom 21/09/08 16:30
14		 desenvolvimento rotina principal	28 hrs	Ter 23/09/08 21:30	Seg 29/09/08 23:00
15		desenvolvimento de rotina de testes de LCD	2 hrs	Ter 23/09/08 21:30	Qua 24/09/08 20:30
16		desenvolvimento rotina de interrupção da serial	8 hrs	Ter 23/09/08 21:30	Sex 26/09/08 20:30
17		desenvolvimento rotina de busca no vetor de produtos/preços	8 hrs	Sex 26/09/08 20:30	Sáb 27/09/08 19:30
18		mostrar no LCD o resultado da busca	4 hrs	Sáb 27/09/08 19:30	Dom 28/09/08 15:00
19		desenvolvimento de interrupção de teclado (rolar barra de rolag	8 hrs	Dom 28/09/08 15:00	Seg 29/09/08 23:00
20		 desenvolvimento exportação USB	8 hrs	Qua 24/09/08 20:30	Sex 26/09/08 22:30
21		desenvolvimento da rotina de tratamento da interrupção do PenD	8 hrs	Qua 24/09/08 20:30	Sex 26/09/08 22:30
22		 Confecção da placa de circuito impressa	54 hrs	Ter 30/09/08 20:00	Dom 12/10/08 18:00
23		Desenho da placa	8 hrs	Ter 30/09/08 20:00	Qui 02/10/08 22:00
24		enviar desenho para empresa para confecção da placa	40 hrs	Qui 02/10/08 22:00	Dom 12/10/08 10:00
25		teste de placa	4 hrs	Dom 12/10/08 10:00	Dom 12/10/08 16:00
26		montagem dos componentes eletronicos na placa	2 hrs	Dom 12/10/08 16:00	Dom 12/10/08 18:00
27		 Controle do Projeto	149,5 hrs	Sex 05/09/08 20:00	Sex 10/10/08 22:00
28		Controle do Projeto 1	2 hrs	Sex 05/09/08 20:00	Sex 05/09/08 22:00
29		Controle do Projeto 2	2 hrs	Sex 12/09/08 20:00	Sex 12/09/08 22:00
30		Controle do Projeto 3	2 hrs	Sex 19/09/08 20:00	Sex 19/09/08 22:00
31		Controle do Projeto 4	2 hrs	Sex 26/09/08 20:00	Sex 26/09/08 22:00
32		Controle do Projeto 5	2 hrs	Sex 03/10/08 20:00	Sex 03/10/08 22:00
33		Controle do Projeto 6	2 hrs	Sex 10/10/08 20:00	Sex 10/10/08 22:00

Figura Apêndice 1.1 – Cronograma do projeto. Modo planilha. Fonte: Do Autor.

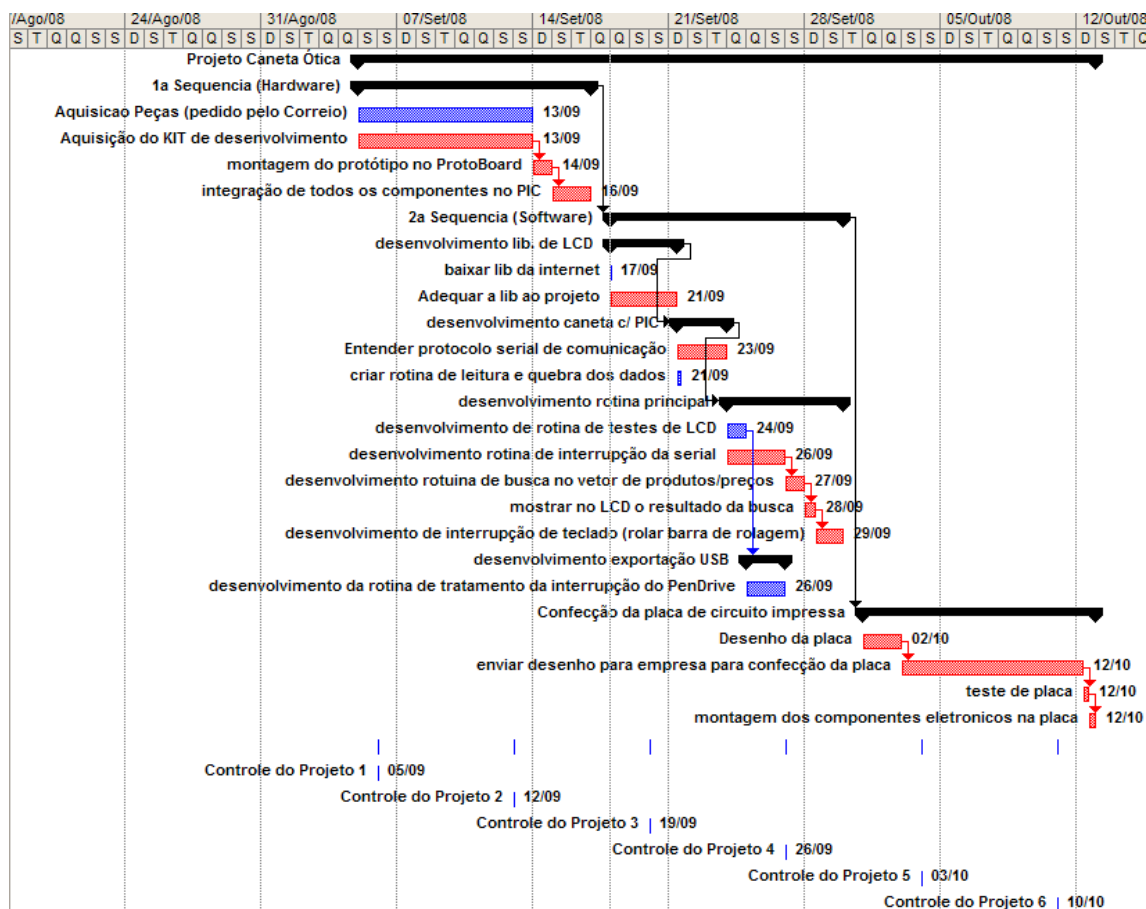
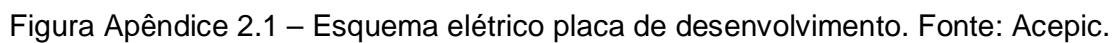


Figura Apêndice 1.2 – Cronograma do projeto. Modo Gráfico de Gannt. Fonte: Do Autor.

Apêndice II

Esquemas da placa de testes do projeto:

Os esquemas a seguir ilustram as conexões de comunicações dos periféricos da placa de testes Acepic, utilizada para testes no projeto.



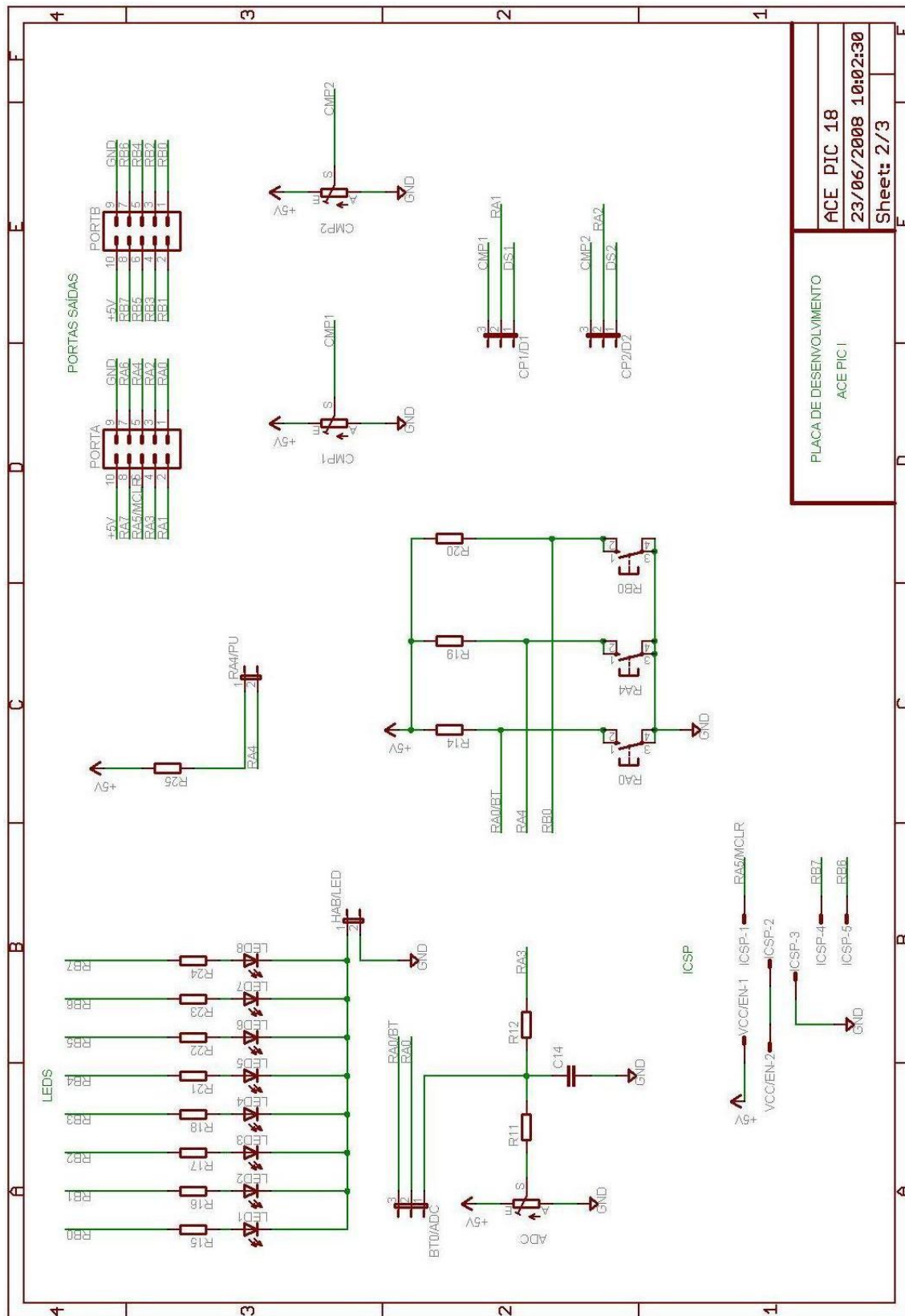


Figura Apêndice 2.2 – Esquema elétrico placa de desenvolvimento. Fonte: Acepic.

Apêndice III

Código-fonte do projeto Caneta Ótica que Registra e Contabiliza os Produtos Automaticamente.

```
#include "LCD4B.c"
char rec[3];

unsigned short asciiprodutos[30] = {0x41,0x43,0x55,0x43,0x41,0x52,0x2E,0x2E,0x2E, // a , c , u , c
,a , r
                                0x46,0x45,0x49,0x4A,0x41,0x4F,0x2E,0x2E,0x2E, // f , e , i , j , a , o
                                0x53,0x41,0x4C,0x2E,0x2E,0x2E,0x2E,0x2E,0x2E};          // s , a , l

//
unsigned char *produto[]= {"Acucar", "Feijao", "Sal", "Produto Invalido", " ", " ", " "};
unsigned char *preco[] = {"1.77", "2.45", "2.99"};
//
unsigned char *msg1, *msg2;
unsigned short linha, visor_inicio=0;
int vlr = 0;
unsigned short QtdeProd = 0;
short retira = 0;
//-----
// funcao para iniciar o prorgama e limpar a memoria
//-----
void Tela_inicio()
{
    LCD_Write(2,3,"Caneta Otica");
    LCD_Write(3,4,"Versao:2.8");
    visor_inicio = 1;
}
//-----
// funcao para escrever os dados para a PenDrive
//-----
void Nome_Arquivo_Serial() //LISTA.TXT
{
    Usart_Write(0x4C);    //L

    Usart_Write(0x69);    //I
    Usart_Write(0x73);    //S
    Usart_Write(0x74);    //T
    Usart_Write(0x41);    //A

    Usart_Write(0x2E);    //.
    Usart_Write(0x74);    //T
    Usart_Write(0x78);    //X
    Usart_Write(0x74);    //T
    //

}
//-----
```

```

// IdxProd : código do produto na memória
// IdxPosicaoInicial -> IdxPosicaoFinal : código do nome do produto em ASCII no vetor asciiprodutos
// QuantidadeDePontos -> Espaços em branco ou ponto
//-----
void Escrever_Produto_PenBS(unsigned short IdxProd,
                           unsigned short IdxPosicaoInicial,
                           unsigned short IdxPosicaoFinal)
{
    unsigned short x;
    //
    for (x=IdxPosicaoInicial; x < IdxPosicaoFinal; x++)
    {
        Usart_Write( asciiprodutos[x] );
    }

    //
    x = EEprom_Read(IdxProd);
    //
    x = ( ( x == 0xFF ) ? 0 : x );
    x = x + 0x30; // converto para ascii
    Usart_Write( x );
    Usart_Write( 0x0D );
}
//-----
//-----
void Esc_Caracter_Serial()
{
    Escrever_Produto_PenBS(0,0,9); //envia acucar definitivamente a PenDrive
    //
    Escrever_Produto_PenBS(1,9,18); //feijao definitivamente a PenDrive
    //
    Escrever_Produto_PenBS(2,18,27); //sal definitivamente a PenDrive
}
//-----
//-----
void Transmitir_Dados_Serial() //LISTA.TXT
{
    //unsigned short vlr;

    // COMANDO PARA ENVIAR DADOS
    Usart_Write(0x08);
    Usart_Write(0x20);
    // COMANDO PARA DIZER O TAMANHO DOS DADOS A SEREM ENVIADOS
    Usart_Write(0x00);
    Usart_Write(0x00);
    Usart_Write(0x00);
    Usart_Write(0x21);
    // COMANDO PARA FECHAR O COMANDO ANTERIOR
    Usart_Write(0x0D); //T
    // ESCRIVE DEFINITIVAMENTE NA PEN-DRIVE
    Esc_Caracter_Serial();
    //
    Usart_Write(0x0D); //T
}
//-----
//-----
void Excluir_Arquivo()
{
    Usart_Write(0x07); //

```

```

    Usart_Write(0x20); //
    Nome_Arquivo_Serial();
    Usart_Write(0x0D); //
}
//-----
// Funcao Para Extrair os dados para a PenDrive
//-----
void PenDrive()
{
    Usart_Write(0x10); //
    Usart_Write(0x0D); //Habilita escrita na forma de comandos resumidos
    //
    Excluir_Arquivo(); // Se o arquivo existir n PenBS ele será excluído
    //
    Usart_Write(0x09); // Comando para abrir o arquivo, crio se ele não existir
    Usart_Write(0x20); //
    //
    Nome_Arquivo_Serial();
    Usart_Write(0x0D); //Comando */
    //
    Transmitir_Dados_Serial();
    //
    // COMANDO PARA FECHAR ARQUIVO
    //
    Usart_Write(0x0A); //
    Usart_Write(0x20); //
    //
    Nome_Arquivo_Serial();
    Usart_Write(0x0D); //Comando */
}
//-----
//-----
void Trata_Produto(int posicaoMemoria, int preco)
{
    unsigned short qtde;

    // captura o
    qtde = EEProm_Read(posicaoMemoria);

    // se deve retirar o produto, subtrai o valor e limpa a memoria
    if ( retira == 1 )
    {
        if ( qtde == 1 || qtde == 0xFF ) {
            msg1 = produto[4];
            msg2 = produto[5];
        }
        if ( qtde != 0xFF )
        {
            if ( qtde == 1 )
                qtde = 0xFF ;
            else
                qtde = qtde - 1;
            vlr = vlr - preco;
            EEProm_Write(posicaoMemoria,qtde);
        }
        retira = 0;
    }
    // senao adiciona o valor e guarda o produto na memoria
    else
    {

```



```

    qtde = ( qtde == 0xFF ) ? 1 : qtde + 1;
    vlr = vlr + preco;
    EEPROM_Write(posicaoMemoria,qtde);
}

// grava a qtde de produtos a ser mostrado
if ( qtde == 0xFF )
    QtdeProd = 0 ;
else
    QtdeProd = qtde;
}
//-----
// Funcao para Tratar toda a regra de negocio do programa Caneta Otica
// analisa se o produto deve ser inserido ou retirado
// escreve no visor LCD
//-----
void Trata_Serial()
{
    int vlrp;

    // se veio lixo da serial pela PenDrive, sai fora
    if (rec[1]=='V')
        return;

    // primeiro produto lido diferente da msg de PENDrive conectada
    if (visor_inicio==1 )
    {
        LCD_Clr();
        Delay_ms(10);
        visor_inicio = 0;
    }

    if ((rec[0]=='3') && (rec[1]=='E'))
    {
        msg1 = produto[0];/"Acucar";
        msg2 = preco [0];
        linha = 1;
        Trata_Produto(0, 177);
    }
    else
    if ((rec[0]=='3') && (rec[1]=='D'))
    {
        msg1 = produto[1];/"feijao";
        msg2 = preco [1];
        linha = 2;
        Trata_Produto(1, 245);
    }
    else
    if ((rec[0]=='3') && (rec[1]=='F'))
    {
        msg1 = produto[2];/"Sal";
        msg2 = preco [2];
        linha = 3;
        Trata_Produto(2, 299);
    }
    else
    {
        linha = 0;
    }
}

```

```

// se leu um dos produtos registrados
if (linha > 0)
{
    // se retira o produto, limpa a linha do produto
    LCD_Write(linha,1 ,msg1);
    LCD_Write(linha,13,msg2);
    if ( QtdeProd )
        LCD_Ch(linha,9, QtdeProd+0x30);

    LCD_Write(4,1,"Valor:  ");
    vlrp = vlr / 1000;
    LCD_Ch(4,12,vlrp+0x30);
    vlrp = (vlr / 100) % 10;
    LCD_Ch(4,13,vlrp+0x30);
    LCD_Ch(4,14,'. ');
    vlrp = (vlr / 10) % 10;
    LCD_Ch(4,15,vlrp+0x30);
    vlrp = vlr % 10;
    LCD_Ch(4,16,vlrp+0x30);
}
else
    LCD_Write(4,1 ,"Produto Invalido");
}
//-----
//-----
void main()
{
    unsigned int i=0;

    PORTA = 0;          //PORTA todo em 0

    TRISA.F2 = 1;        //Opcoes para ativar a chamada dos botoes na placa
    TRISA.F3 = 1;        //RA4 configurado para entrada
    TRISA.F4 = 1;
    TRISA.F5 = 1;

    CMCON = 0x07;

    LCD_INI();           //LCD_INI porque está no padrão do MicroC

    Tela_inicio();

    // inicializ a porta serial com a caneta optica
    Usart_Init(9600);

    while(1)
    {
        // Exclusao de produtos
        if (!PORTA.F3)
        {
            LCD_Ch(4,1,'A');
            LCD_Ch(4,2,'P');
            LCD_Ch(4,3,'A');
            LCD_Ch(4,4,'G');
            LCD_Ch(4,5,'A');
            LCD_Ch(4,6,'R');
            retira = 1;
        }
    }
}

```

```

if (!PORTA.F4)
{
    LCD_Clr();
    //LCD_Write(2,2,"APAGANDO DADOS..");
    Delay_ms(500);
    // limpando a regioao de memoria
    for (i=0;i<3;i++)
    {
        EEProm_Write(i,0xFF);
        Delay_ms(10);
    }

    Tela_inicio();
    //i=0;
    vlr = 0;
}

if(!PORTA.F5)
{
    LCD_Clr();
    PenDrive();
}

if(usart_data_ready()) //Retorna 1 se existe dados ou 0 se não existe
{
    i = 0;

    while ( usart_data_ready() )
    {
        rec[i] = usart_read();
        i = i + 1;
        Delay_ms(250);
    }

    rec[2] = '\0';
    Trata_Serial();

}

}
//-----

```